SNP Retrieval and Manipulation

Daniel M. Gatti

July 2, 2013

1 Introduction

This vignette demonstrates how to use the SNP manipulation functions to retrieve, subset and plot SNPs from large data sets. There are several large SNP data sets, such as the Sanger Mouse Genomes and a set of imputed SNPs from the University of North Carolina at Chapel Hill. The vignette is oriented toward the researcher that has mapped a QTL to a given locus and wants to plot genes and SNPs in the locus.

Due to the size of the SNP files, it is inadvisable to read the entire file into memory in order to retrieve a subset of SNPs. Rather, the SNP file are zipped and indexed to reduce file size and speed up SNP retrieval.

2 Installation and Required Files

SNPtools depends upon four other R packages that should be installed beforehand: IRanges GenomicRanges Biostrings Rsamtools

To install these, you can run:

```
source("http://bioconductor.org/biocLite.R")
biocLite(c("IRanges", "GenomicRanges", "Biostrings", "Rsamtools"))
```

You will need a SNP file and a file with gene locations. These can be obtained here. You will need both the SNP files as well as the gene location files. We use gene locations from The Jackson Laboratory Mouse Genome Informatics. Download both the *.gz file and the *.gz.tbi file. The files are zipped and indexed using Tabix. The *.gz file contains the data and the *.gz.tbi file contains the index to the data.

To load the library, use:

```
> library(SNPtools)
```

3 Retrieving Sanger SNPs

We have taken the Sanger SNPs and reformatted them into a text format, zipped and indexed using Tabix. Use of this data should include the follwing references: Keane TM, Goodstadt L, Danecek P, et al. (2011) Mouse genomic variation and its effect on phenotypes and gene regulation, *Nature*, 477(7364):289-294, Yalcin

B, Wong K, Agam A, et al. (2011) Sequence-based characterization of structural variation in the mouse genome, *Nature*, 477(7364):326-329. The available strains are:

129P2/OlaHsd	129S1/SvImJ	129S5SvEvBrd
A/J	AKR/J	BALB/cJ
C3H/HeJ	C57BL/6J	C57BL/6NJ
CAST/EiJ	CBA/J	DBA/2J
FVB/NJ	LP/J	NOD/ShiLtJ
NZO/HlLtJ	PWK/PhJ	WSB/EiJ

The allele calls have a qulaity score of either 0 or 1 corresponding to bad or good, respectively. SNPs can be subset by genomic region, strains, polymorphic status and quality.

3.1 Making a SNP Plot with a single command

There is a single function that will perform all of the SNP analysis detail below. The commands in the next section break the work up into separate pieces, which may be useful when you are making mulitple plots or when you want more control over your data.

First set the path to the SNP file and the MGI file that you downloaded. There are two files with each download. Select the one that ends in "gz", not "tbi". You can download the files from SNPtools and set them to paths in your local file system. Below, we set them to web URLs at the Jackson Laboratory.

```
> snp.file = "http://cgd.jax.org/tools/SNPtools/Build38/sanger.snps.NCBI38.txt.gz"
> mgi.file = "http://cgd.jax.org/tools/SNPtools/MGI/MGI.20130305.sorted.txt.gz"
```

Next, get the strain names that are available in the SNP file. If you are mapping with the Collaborative Cross or Diversity Outbred founders, then you can skip this argument. They are the default strains. Otherwise, it's a good idea to select them directly from the list of available strains because there are small differences in nomenclature that can cause mismatches. Here, we just get the CC founders to show how to retreive them.

```
> available.strains = get.strains(snp.file)
> strains = available.strains[c(4, 2, 8,  15, 16, 10, 17, 18)]
> strains

[1] "A/J"         "129S1/SvImJ" "C57BL/6J"    "NOD/ShiLtJ"  "NZO/HlLtJ"
[6] " CAST/EiJ"   "PWK/PhJ"     "WSB/EiJ"
```

You can also load in your QTL data. The format requires a data.frame with three columns containing chromosome, Mb position and LOD score. See the example below.

```
> data(qtl)
> head(qtl)

  Chromosome Position       lod
1          1  3252796 0.9701113
2          1  3336839 0.9658992
```

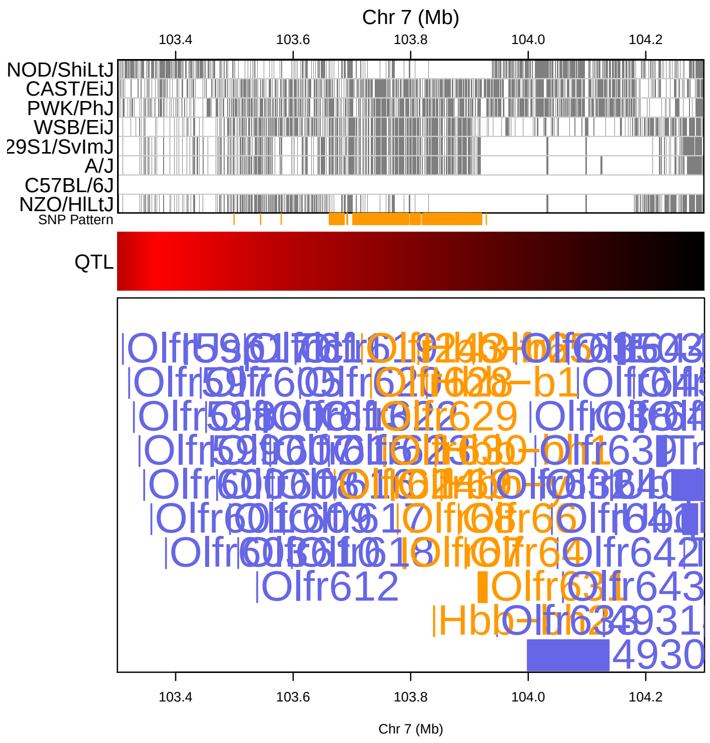
```
3           1  3668628 0.9190987
4           1  3977130 0.8604857
5           1  4430623 0.8551462
6           1  4531029 0.7322654
```

This QTL data is for mean corpuscular hemoglobin concentration.

Then run `variant.plot`, which returns a list of all SNPs that match the strain pattern, along with the genes that the lie within. We know from prior information that C57BL/6J, NOD/ShiLtJ and NZO/HlLtJ have one allele and the remaining five strains have the other.

```
> cat.snps = variant.plot(var.file = snp.file, mgi.file = mgi.file,
+            chr = 7, start = 103.3, end = 104.3, strains = strains,
+            pattern = strains[c(3:5)], qtl = qtl)
```

```
[1] "Checking arguments..."
[1] "Retreiving Variants..."
[1] "25450 Variants in region."
[1] "Getting SNPs that match the pattern..."
[1] "2751 variants fit the pattern."
[1] "Categorizing pattern variants..."
[1] "19 genes have variants that fit the pattern between the 3'UTR and the 5'UTR (includin
[1] "Getting genes in region..."
[1] "66 genes in region."
[1] "Drawing plot..."
```



The function returns the SNPS that match the desired allele pattern along with information about whether they occur in exons. As you can see, the SNPs that match the pattern cluster over a set of hemoglobin (*Hbb*) genes. These are certainly plausible candidate for a hemoglobin related trait.

```
> head(cat.snps)
```

```
               ID CHROM       POS REF ALT symbol   id       type
5037   rs49176439     7 103499458   C   A   <NA> <NA> intergenic
6312            .     7 103544588   G A,C   <NA> <NA> intergenic
7199            .     7 103579635   T   C   <NA> <NA> intergenic
9107   rs45805035     7 103661698   T   C   <NA> <NA> intergenic
9110   rs47660696     7 103661709   A   G   <NA> <NA> intergenic
9111  rs215837914     7 103661720   G   T   <NA> <NA> intergenic
```

4 Retrieving and Plotting SNPs for a QTL Interval

In Peters et.al, Blood, 2010, the authors found a QTL for mean corpuscular hemoglobin concentration on Chr 7. Through a variety of analyses, the authors narrowed the QTL interval to 108.5 - 111.58 Mb. Below is a table of strains that contribute high and low alleles.

High Allele	Low Allele
129/SvImJ	C57BL/6J
A/J	NOD/ShiLtJ
WSB/EiJ	NZO/HlLtJ
CAST/EiJ	
PWK/PhJ	

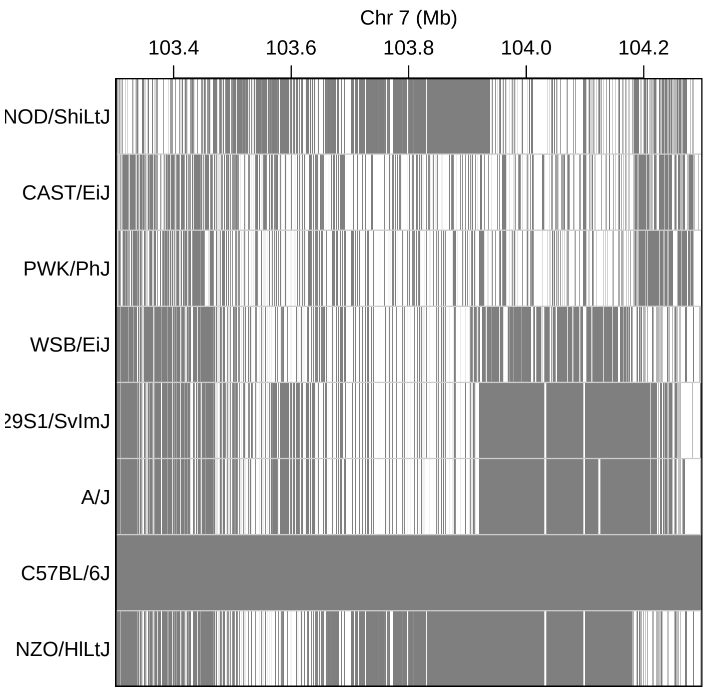
First, retrieve the polymorphic SNPs for your strains in the QTL interval (Chr 7: 103.3 - 104.3 Mb).

```
> available.strains = get.strains(snp.file)
> strains = available.strains[c(4, 2, 8,  15, 16, 10, 17, 18)]
> snps = get.variants(chr = 7, start = 103.3, end = 104.3,
+        strains = strains, type = "snp")
> nrow(snps)
```

```
[1] 25394
```

There are 25,394 SNPs in this region. Plot the alleles, using C57BL/6J as the reference.

```
> nsnps = convert.variants.to.numeric(snps)
> snp.plot(nsnps, ref = "C57BL/6J", cluster = T)
```



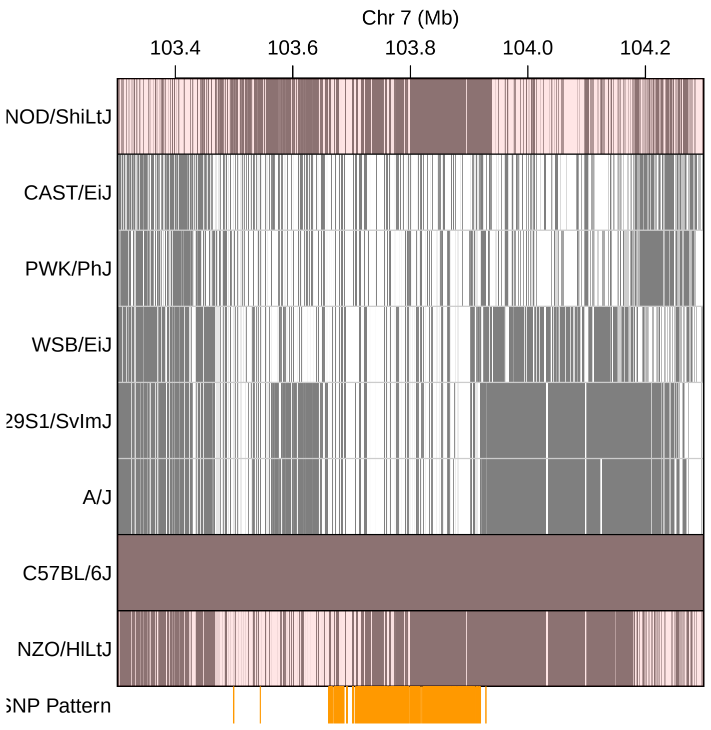
Since C57BL/6J, NOD/ShiLtJ and NZO/HlLTJ have low CHCM values and the remaining strains have higher values, candidate SNPs will have the same allele in C57BL/6J and SM/J, but a different allele for the remaining strains. Use `get.pattern.snps` to retrieve the SNPs that fit this pattern.

```
> strain.subset = c("C57BL/6J", "NOD/ShiLtJ", "NZO/HlLtJ")
> snp.subset = get.pattern.variants(snps, strain.subset)
> nrow(snp.subset)
```

```
[1] 2596
```

This has narrowed the SNPs down to 2,596. Next, plot the SNPs in the QTL interval, marking the SNPs that fit this pattern, highlighting the low allele strains.

```
> snp.plot(nsnps, ref = "C57BL/6J", cluster = T, highlight = strain.subset,
+          pattern = snp.subset)
```

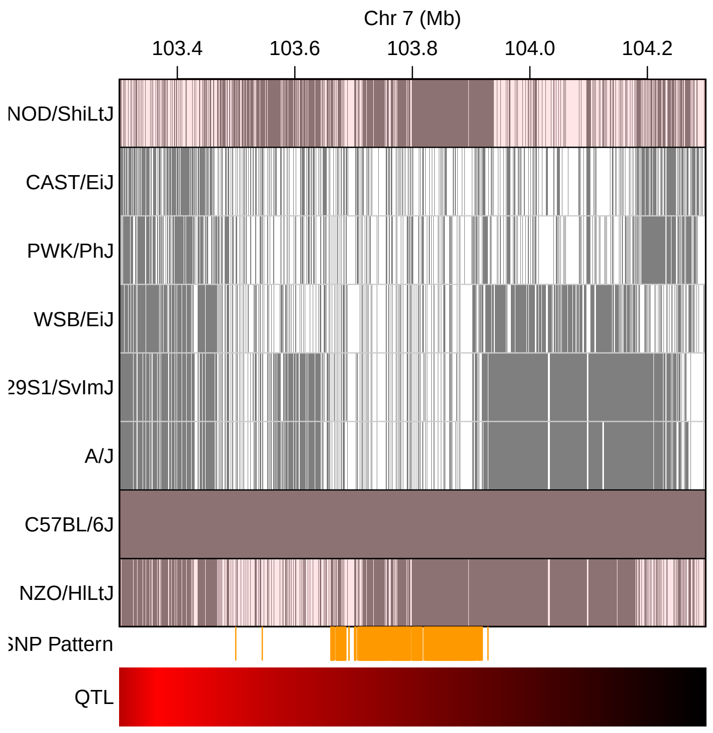


If you have a LOD score (or other QTL statistic), you can add it to the SNP plot as a heat map. Start by loading in your QTL data. It must be a data.frame with three columns containing the chromosome, bp position and QTL statistic (such as the LOD) for the region you are plotting. Then use the `qtl` argument to feed it into `snp.plot`.

```
> data(qtl)
> head(qtl)

  Chromosome Position       lod
1          1  3252796 0.9701113
2          1  3336839 0.9658992
3          1  3668628 0.9190987
4          1  3977130 0.8604857
5          1  4430623 0.8551462
6          1  4531029 0.7322654

> snp.plot(nsnps, ref = "C57BL/6J", cluster = T, highlight = strain.subset,
+          pattern = snp.subset, qtl = qtl)
```

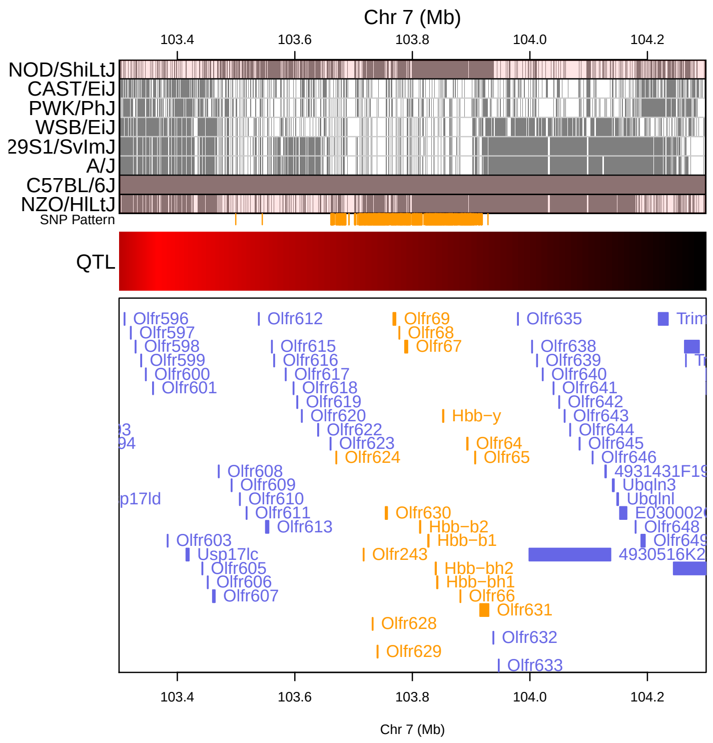


The LOD score is highest in a region around 103.4 Mb. The gene could be anywhere in the QTL interval, but we often start where the peak is highest and work outwards. Further, although SNPs that fit the allele pattern occur throughout the QTL interval, they seem to primarily cluster in a region between 103.6 and 103.9 Mb. Note that you should not exclude the other regions completely, but we will focus on the largest cluster first.

4.1 Retrieving Mouse Genome Informatics (MGI) features

Next, we will add gene locations to the plot. If you have the MGI feature file in a parsed and Tabix indexed format, you can use the`get.mgi.features()` function to add these to your allele plot. Note that `get.mgi.features()` has a file argument like `get.variants` and that you can specify an alternate file location.

```
> mgi = get.mgi.features(chr = 7, start = 103, end = 105,
+        source = "MGI", type = "gene")
> mgi = mgi[-grep("^Gm", mgi$Name),]
> snp.plot(nsnps, ref = "C57BL/6J", cluster = T,highlight =
+           strain.subset, pattern = snp.subset, qtl = qtl,
+           mgi = mgi)
```



As you can see, there is a cluster of hemoglobin (Hbb) genes in the middle of this region. The three low allele strains all share an allele pattern that differs from the high allele strains. This matches the conclusion of Peters et.al., who found that there is paralogous copy number variation of the Hbb genes at this locus.

5 Intersecting SNPs and Genes

Once you have a set of candidate SNPs, you may wish to prioritize them by intersecting them with genes. The `categorize.variants()` function will classify SNPs according to whether they lie within exons, introns or the 3' or 5' untranslated region(UTR).

```
> snps = get.variants(chr = 7, start = 103.3, end = 104.3,
+          strains = strains, type = "snp")
> snp.type = categorize.variants(snps)
> head(snp.type[grep("Hbb", snp.type$symbol),])
```

```
               ID CHROM       POS REF ALT symbol   id   type
13180  rs50940139     7 103812594   A   C Hbb-b2 <NA>   3UTR
13181 rs253503843     7 103812638   C   A Hbb-b2 <NA>   3UTR
13182           .     7 103812647   A   G Hbb-b2 <NA>   3UTR
13183 rs236944093     7 103812659   G   A Hbb-b2 <NA> intron
13184  rs50625643     7 103812682   C   T Hbb-b2 <NA> intron
13185  rs50259061     7 103812683   A   G Hbb-b2 <NA> intron
```

6 Retrieving Indels and structural variants

The `get.variants` function can also retreive Indels and structural variants. You must supply the file argument as well as the type argument. Here, we use a file with only Collaborative Cross/Diversity Outbred founders.

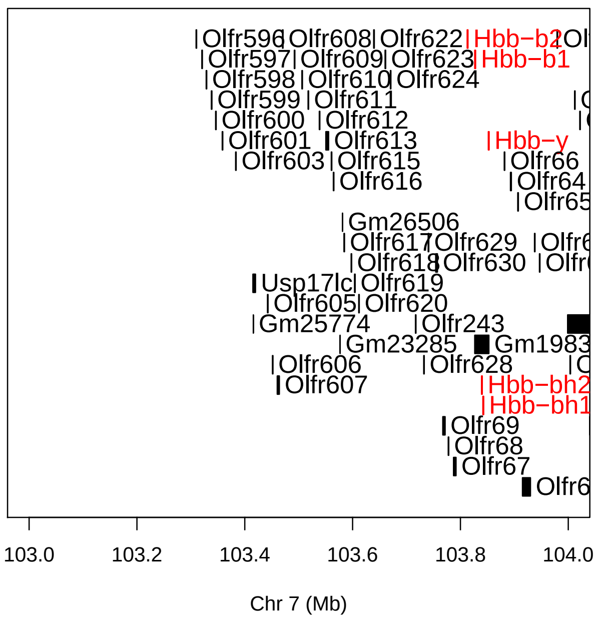
```
> indels = get.variants(file = "http://cgd.jax.org/tools/SNPtools/Build38/cc.indels.NCBI38
+        chr = 7, start = 103.3, end = 104.3, type = "indel")
> head(indels)
```

```
           ID CHROM       POS          REF         ALT
1           .     7 103302138            T        TCAA
2           .     7 103302229           CA           C
3           .     7 103302998 TTGGTTTTTGTA           T
4           .     7 103302999  TGGTTTTTGTA TTTTTTGTA,T
5           .     7 103303051        GAATT           G
6 rs230506517     7 103303192           TG           T
                        A/J quality                 C57BL/6J quality
1                       T/T       1                      T/T       1
2                     CA/CA       1                    CA/CA       1
3 TTGGTTTTTGTA/TTGGTTTTTGTA       1 TTGGTTTTTGTA/TTGGTTTTTGTA       1
4   TGGTTTTTGTA/TGGTTTTTGTA       1   TGGTTTTTGTA/TGGTTTTTGTA       1
5               GAATT/GAATT       1              GAATT/GAATT       1
6                     TG/TG       1                    TG/TG       1
               129S1/SvImJ quality NOD/ShiLtJ quality
1                       T/T       1  TCAA/TCAA       1
2                     CA/CA       1      CA/CA       1
3 TTGGTTTTTGTA/TTGGTTTTTGTA       1        T/T       1
4   TGGTTTTTGTA/TGGTTTTTGTA       1        T/T       0
5               GAATT/GAATT       1        G/G       1
6                     TG/TG       1      TG/TG       1
                 NZO/HlLtJ quality                  CAST/EiJ quality
1                       T/T       1                TCAA/TCAA       1
2                     CA/CA       1                      C/C       1
3 TTGGTTTTTGTA/TTGGTTTTTGTA       1 TTGGTTTTTGTA/TTGGTTTTTGTA       1
4   TGGTTTTTGTA/TGGTTTTTGTA       1   TGGTTTTTGTA/TGGTTTTTGTA       1
5               GAATT/GAATT       1              GAATT/GAATT       1
6                     TG/TG       1                      T/T       1
                   PWK/PhJ quality                   WSB/EiJ quality
1                 TCAA/TCAA       1                      T/T       1
2                     CA/CA       1                    CA/CA       1
3 TTGGTTTTTGTA/TTGGTTTTTGTA       1 TTGGTTTTTGTA/TTGGTTTTTGTA       1
4   TGGTTTTTGTA/TGGTTTTTGTA       1      TTTTTTGTA/TTTTTTGTA       1
5               GAATT/GAATT       1              GAATT/GAATT       1
6                     TG/TG       1                    TG/TG       1
```

7 Plotting Gene Locations

The function `gene.plot` plots gene locations with labels to the right. To use it, retrieve MGI genes and feed them into the `mgi` argument. Optionally, you make select some genes to highlight.

```
> mgi = get.mgi.features(chr = 7, start = 103.3, end = 104.3,
+        source = "MGI", type = "gene")
> col = rep(1, nrow(mgi))
> col[grep("Hbb", mgi$Name)] = 2
> gp = gene.plot(mgi, col = col)
```



8 Variant File Format for SNPtools

While the VCF format is useful, it requires too much time to parse these files in R. We convert the large VCF files into text and index them with Tabix. Parsing these files for plotting is much faster.

Since the files are indexed with Tabix, they must be tab delimited. For SNPs and Indels, the first five columns are called ID, CHROM, POS, REF and ALT. They contain the SNP ID (if available), the chromosome, the base pair position, the reference allele and the alternate alleles, separated by commas. There follows two columns per strain. The first column must be named for the strain and contain a pair of allele (i.e. AA, CC, AC, etc.). The next column must be called quality and contains a 0 or 1, which corresponds to the FI tag from the Sanger VCF file. 0 is bad and 1 is good. If you do not have quality information for your SNPs, then code all of the quality columns as 1.

ID	CHROM	POS	REF	ALT	129P2/OlaHsd	quality
rs261849515	7	103000927	G	A	GG	1
.	7	103001931	G	C,A	GG	1
.	7	103001952	A	T	AA	1
.	7	103002096	A	C	AA	1
.	7	103002180	G	A	GG	1
.	7	103002245	C	T	CC	1