SoyaGen 2017 Haplotype Workshop

Marc-André Lemay

2017-12-19

Context

The package `autohaplo` has been developed to allow the definition of haplotypes based on a gene-centered approach. The package automates the haplotype-definition approach described by Tardivel et al. (The Plant Genome 2014).

This workshop aims at presenting the basics of the approach to potential users of the package and provide them with the necessary autonomy for carrying their own haplotype analyses. The development of the package `autohaplo` is an ungoing project. Documentation for most functions, though not all, is already available. New functionalities are also likely to be added over time. For the moment, the package is only available through the authors (please contact marc-andre.lemay.2@ulaval.ca), but should available from the web over the course of 2018.

Overview of the approach

The haplotype definition approach of `autohaplo` is gene-centered, that is, the package defines haplotypes that represent putative alleles of genes of interest based on the position of the gene. `autohaplo` essentially looks at a genomic interval surrounding a gene of interest and selects SNPs that are likely to be useful in defining haplotypes. This selection of SNPs is done in three steps (filtering, clustering and final selection) that will be described below.

The two basic inputs for package `autohaplo` are the position of a gene of interest and a genotype file indicating the genotypes of a population of interest at different SNP markers. Providing population structure and/or kinship files is optional, but recommended since these data improve the accuracy of linkage disequilibrium estimates and thus the quality of the analysis.

The computation carried out by `autohaplo` also depends on a large set of parameters that will influence the results and output of the analysis. These parameters will be described below along with the example code.

Filtering

In a first step of the analysis, `autohaplo` filters SNPs based on different criteria. Most importantly, only SNPs located on the chromosome on which the gene is located and at a maximal (physical) distance from the gene center are retained for further processing. The maximal distance to consider will depend on the pattern of linkage disequilibrium (LD) in the region where the gene is found, but distances of a few hundred kb up to 1 Mb will likely be a good first guess.

Other filtering steps are optional since they can be performed externally, prior to analysing the data with `autohaplo`. Including these filtering functionalities in the package ensures that users who want to try different filtering parameters can do so iteratively without having to go back and forth between `R` and other software. At the moment, SNPs can be filtered based on minor allele frequency, proportion of heterozygosity, proportion of missing data and minor allele count.

Clustering

After the initial filtering step, `autohaplo` computes the linkage disequilibrium between all the pairs of markers that have been retained. A clustering is then be performed to reduce the set of markers to a set of informative markers. On each side of the gene, all marker pairs with a linkage disequilibrium coefficient higher than a given threshold are clustered and only one marker per cluster is kept, with the assumption that all markers in a cluster provide the same information.

Final selection of the markers

After the clustering step, a final selection step is performed on the markers retained to keep only those that are considered informative in defining haplotypes in the region of interest. Two markers will be considered informative if they are located on different "sides" (5' vs 3') of the gene center and are in linkage disequilibrium with each other. This is based on the assumption that markers that are in linkage disequilibrium across the gene center are likely to also be in linkage disequilibrium with genetic polymorphisms located inside the gene sequence.

Example code with gene PhyA3 of soybean

As an example, we will use the package `autohaplo` to define haplotypes at the PhyA3 maturity gene of soybean. This gene is located on the chromosome 19 of the soybean reference genome.

The genotyping data used for this example have been generated by GBS for 67 soybean lines. The data have not been imputed, thus there are still missing genotypes. For the purpose of this workshop, only the markers on chromosomes 19 and 20 have been kept.

R packages

The following packages are needed for `autohaplo` to function accurately. This step will not be necessary once the package will be released, but for the moment, loading the libraries prior to using the package is necessary to avoid bugs.

```
# Loading the packages required for the analysis
library(autohaplo)
library(snpStats)
library(VariantAnnotation)
library(ggplot2)
library(reshape2)
library(LDcorSV)
library(GenomeInfoDb)
```

Analysis parameters

The analysis parameters are built through a helper function called `haplo_params`. This step separates the specfication of the parameters from the actual computation of the haplotypes, and thus grants more flexibility to the user. The parameters used in this analysis are listed and explained here. The description of other parameters can be found in the documentation of the function.

- `input_file`: name of the .hapmap or .vcf genotype file
- `structure_file`: name of the structure file (optional). See the example structure file for formatting requirements.

- `kinship_file`: name of the kinship file (optional). See the example kinship file for formatting requirements.
- `gene_db_file`: name of a file providing information about genes of interest
- `chr_db_file`: name of the file providing information about the size of the chromosomes
- `gene_name`: name of the gene for which haplotypes are to be defined
- `R2_measure`: R^2 linkage disequilibrium measure to use for the final selection step
- `cluster_R2`: R^2 linkage disequilibrium measure to use for the clustering step. Defaults to the same measure as `R2_measure`
- `max_missing_threshold`: the maximum proportion of missing genotypes allowed for a marker
- `max_het_threshold`: the maximum proportion of heterozygous genotypes allowed for a marker
- `min_alt_threshold`: the minimum frequency of the minor allele for a marker to be retained
- `min_allele_count`: the minimum number of times the minor allele has to be seen for a marker to be retained
- `cluster_threshold`: the minimum R^2 for two markers to be clustered
- `max_flanking_pair_distance`: the maximum distance (in bp) that can separate two markers in linkage disequilibrium at the final selection step
- `max_marker_to_gene_distance`: the maximum distance (in bp) from a marker to the center of the gene of interest
- `marker_independence_threshold`: the minimum R^2 for two markers to be considered in linkage disequilibrium at the final selection step

```
# Storing the parameters of the analysis in an object
params <- haplo_params(input_file = "genotypes.hmp.txt",
                       structure_file = "structure.txt",
                       kinship_file = "kinship.txt",
                       gene_db_file = "gene.db.txt",
                       chr_db_file = "gmax.chr.sizes.txt",
                       gene_name = "PhyA3",
                       R2_measure = "r2vs",
                       cluster_R2 = "r2vs",
                       max_missing_threshold = 0.6,
                       max_het_threshold = 0.05,
                       min_alt_threshold = NULL,
                       min_allele_count = 4,
                       cluster_threshold = 0.8,
                       max_flanking_pair_distance = 500000,
                       max_marker_to_gene_distance = 300000,
                       marker_independence_threshold = 0.6)
```

Computing the haplotypes

The function `haplo_selection` performs the computation that leads to the definition of haplotypes. The function will output some information on the screen while it is running. For this example, you can expect it to run under 10 seconds, but it can take considerably longer on larger datasets. Steps are currently taken to improve the performance of the package.

```
results <- haplo_selection(params)
```

```
## coercing object of mode  numeric  to SnpMatrix
## Total number of markers : 3803
## Markers located on chromosome 19 : 2237
## Markers less than 3e+05 bp from gene center : 51
## Number of biallelic markers : 50
## Markers passing missing data filter : 50
```

```
## Markers passing heterozygosity filter : 50
## No MAF filter applied.
## Markers passing MAC filter : 50
```

Generating the output of the analysis

The results of the call to `haplo_selection` above were stored in an object called `results`, but extracting the information from this object is not straightforward. This is why helper functions were created to generate informative output graphs and files that allow interpreting the results and diagnosing any problems.

The object `graph_list` generated below defines a list of graphs to be generated upon applying the function `autohaplo_output` on the `results` object. The results will be output to a directory called "results". By default, `autohaplo` does not allow overwriting a directory that already exists. If you are to perform the analysis several times, you will therefore need to either delete the results directory after each run or change the name of the output directory.

For every step in the definition of haplotypes using `autohaplo`, the `graph_list` object defines the graphs to be generated. Four types of graphs (`density`, `matrix`, `distance` and `genotypes`) can be generated for each step of the marker selection process, but not all these combinations are useful or make sense. We will have a look at some of these graphs below.

```
graph_list <- list("All_markers" = "density",
                   "Filtered_markers" = c("matrix", "distance", "genotypes"),
                   "Clustered_markers" = c("matrix", "genotypes"),
                   "Selected_clusters" = c("matrix", "genotypes"),
                   "Selected_markers" = c("matrix", "genotypes"),
                   "Haplotypes" = c("genotypes"))

autohaplo_output(results, output_dir = "results", graphs = graph_list)
```

Looking at the output of the analysis

Running the commands above should have created a "results" directory containing all the outputs of the analysis. If this is not the case for any reason, you have been provided with a directory called "expected results" so you can still look at the output of the analysis.

Log file

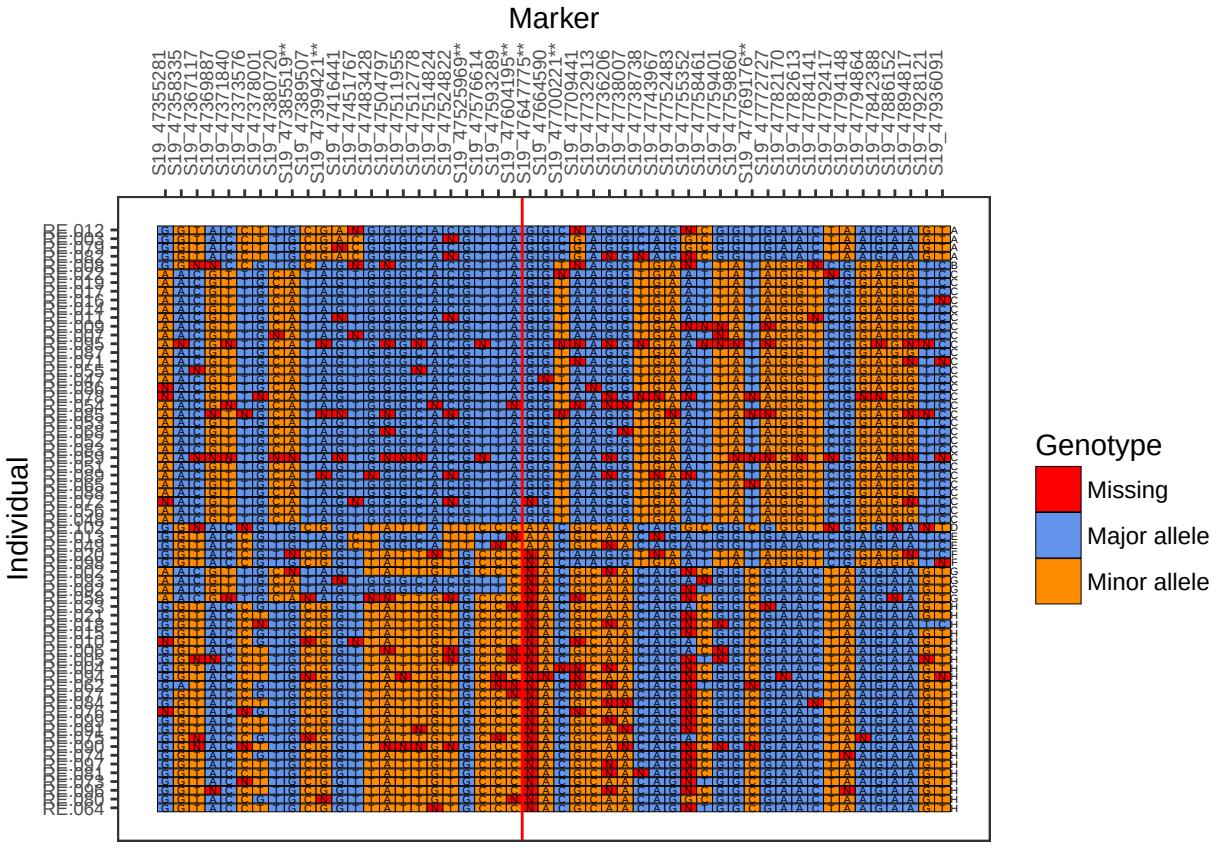
The first file to look at before analyzing the results is the "Log.txt" file. This file gives the user information about the analysis that was performed, the number of distinct haplotypes that were identified, as well as the number of lines corresponding to each of the haplotypes.

```
## Analysis parameters:
##   INPUT FILES
##       Input_file : genotypes.hmp.txt
##       File_format : hapmap
##       Structure_file : structure.txt
##       Kinship_file : kinship.txt
##       Gene_database_file : gene.db.txt
##       Chromosome_database_file : gmax.chr.sizes.txt
##   GENE CHARACTERISTICS
##       Gene_name : PhyA3
##       Gene_chromosome : 19
```

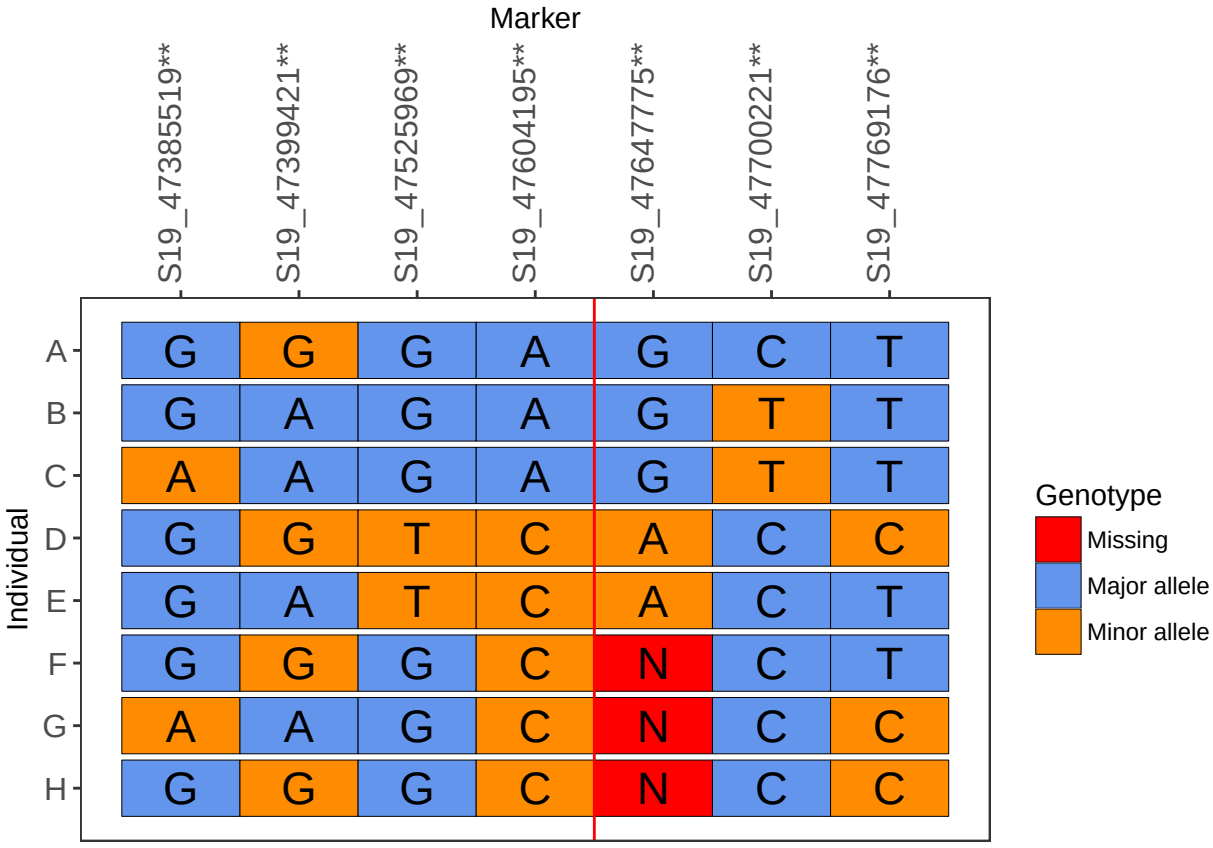
```
##       Gene_start : 47633059
##       Gene_end : 47641958
##       Gene_center : 47637508
##       Cluster_R2_measure : r2vs
##       R2_measure : r2vs
##   FILTERING PARAMETERS
##       Minimum_alternate_allele_frequency : NA
##       Maximum_heterozygous_frequency : 0.05
##       Maximum_missing_allele_frequency : 0.6
##       Minimum_allele_count : 4
##   PAIR SELECTION PARAMETERS
##       Marker_cluster_threshold : 0.8
##       Marker_independence_threshold : 0.6
##       Maximum_flanking_pair_distance : 5e+05
##       Maximum_marker_to_gene_distance : 3e+05
## Marker clustering and filtering results:
##  Total number of markers: 3803
##  Number of markers located on chromosome 19 : 2237
##  Number of markers less than 3e+05 bp from gene center : 51
##  Number of biallelic markers : 50
##  Number of markers passing missing data filter : 50
##  Number of markers passing heterozygosity filter : 50
##  Number of markers passing MAF filter : 50
##  Number of markers passing MAC filter (final number kept for clustering): 50
##  Total markers kept following clustering: 15
## Selecting marker pairs in LD across gene:
##  Total markers kept: 7
##  Haplotype size (distance between two farthest markers): 383657
## Haplotype assignment :
##  Number of distinct haplotypes : 8
##  Number of individuals assigned to haplotype A : 4
##  Number of individuals assigned to haplotype B : 1
##  Number of individuals assigned to haplotype C : 29
##  Number of individuals assigned to haplotype D : 1
##  Number of individuals assigned to haplotype E : 2
##  Number of individuals assigned to haplotype F : 2
##  Number of individuals assigned to haplotype G : 4
##  Number of individuals assigned to haplotype H : 24
##  Number of individuals not unambiguously assigned a haplotype: 0
```

Genotype plots

So-called "genotype plots" allow visualizing the genotypes of the different individuals at the markers that have been kept at a given point of the analysis. For example, the following graph shows the genotypes of the 67 individuals for all of the markers that have been kept following the initial filtering step. In this graph and in all following graphs, the double asterisks denote markers that have been retained to define haplotypes, while the vertical red bar indicates the position of the center of the gene.

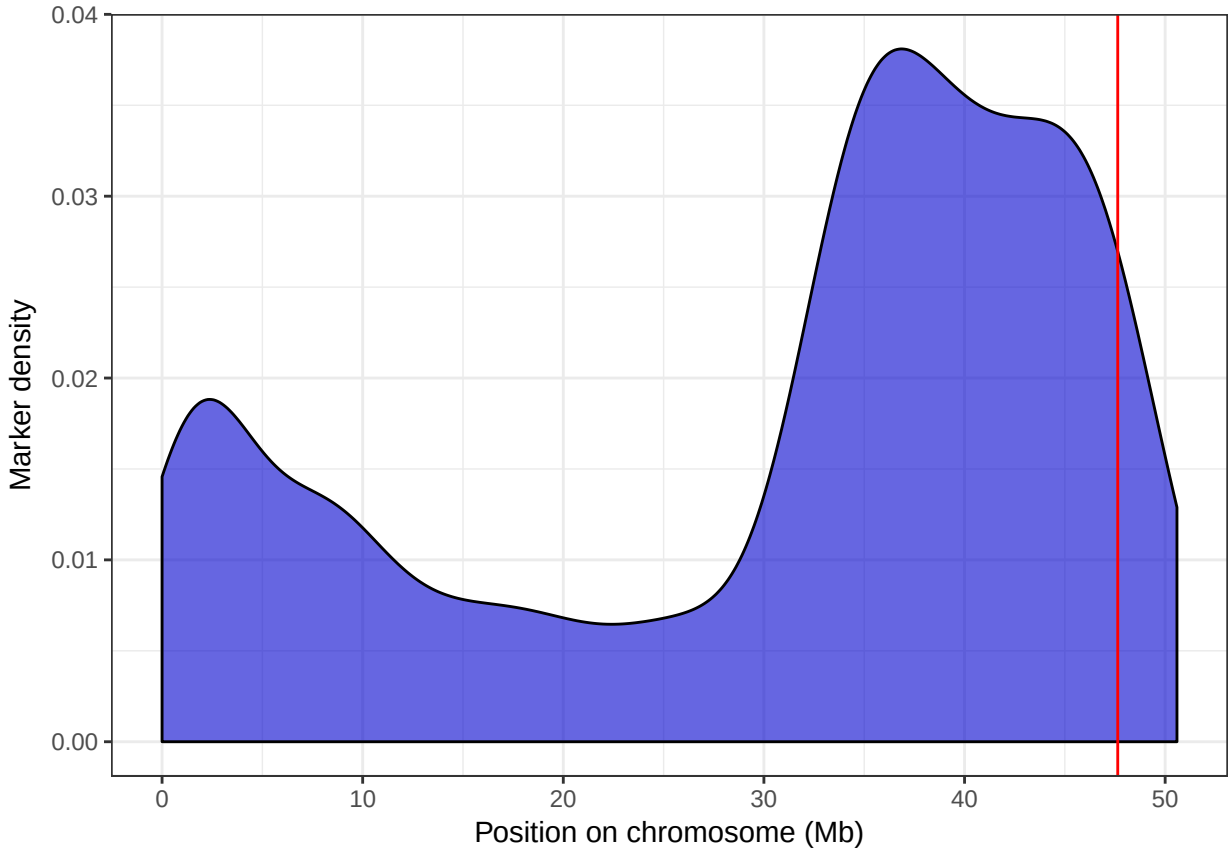


The "genotype plot" of the haplotypes also enables looking at the set of haplotypes defined at the end of the analysis:



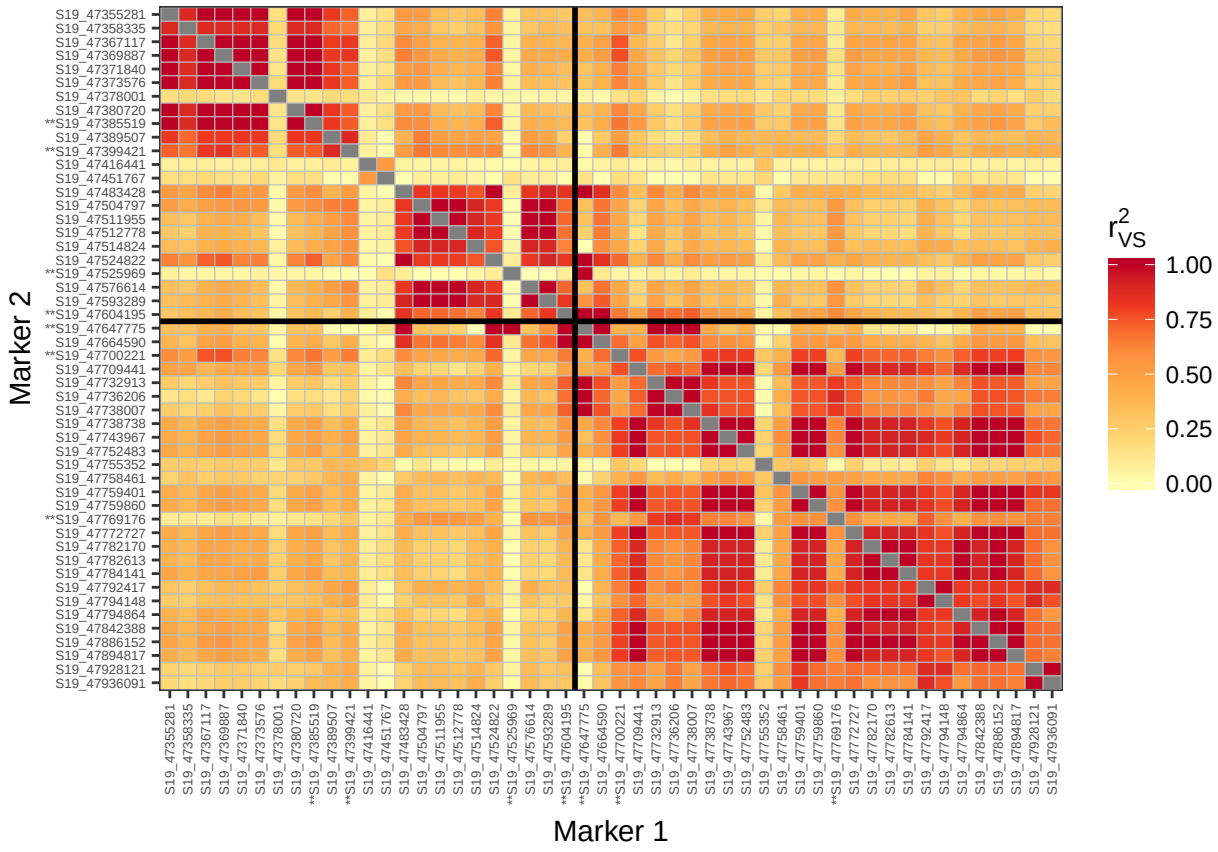
Density plots

The density plot is used to look at the density of markers along the chromosome. It is especially useful to look at the density of markers of the initial set of markers (prior to filtering), as it might indicate a lack of markers surrounding the position of the gene of interest. On the following figure, the red line is located at the center of the PhyA3 gene and suggests that this gene is located in a region with sufficient marker density.

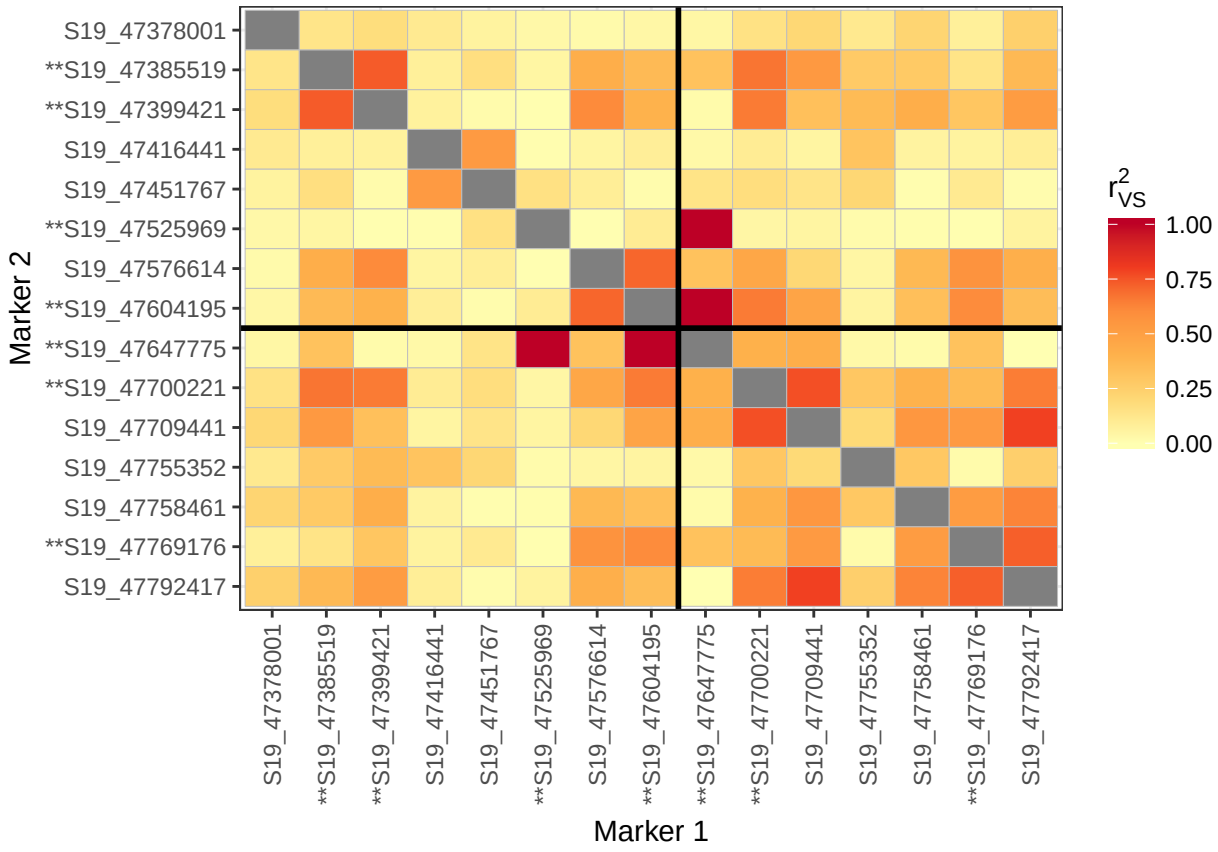


Linkage disequilibrium matrices

These graphs are useful for diagnosing problems in the analysis and understanding the steps implemented in the package. It shows the patterns of linkage disequilibrium (LD) between different markers at different steps of the analysis. The following graph shows the pattern of LD between all markers after filtering, but before clustering. It shows obvious LD blocks:

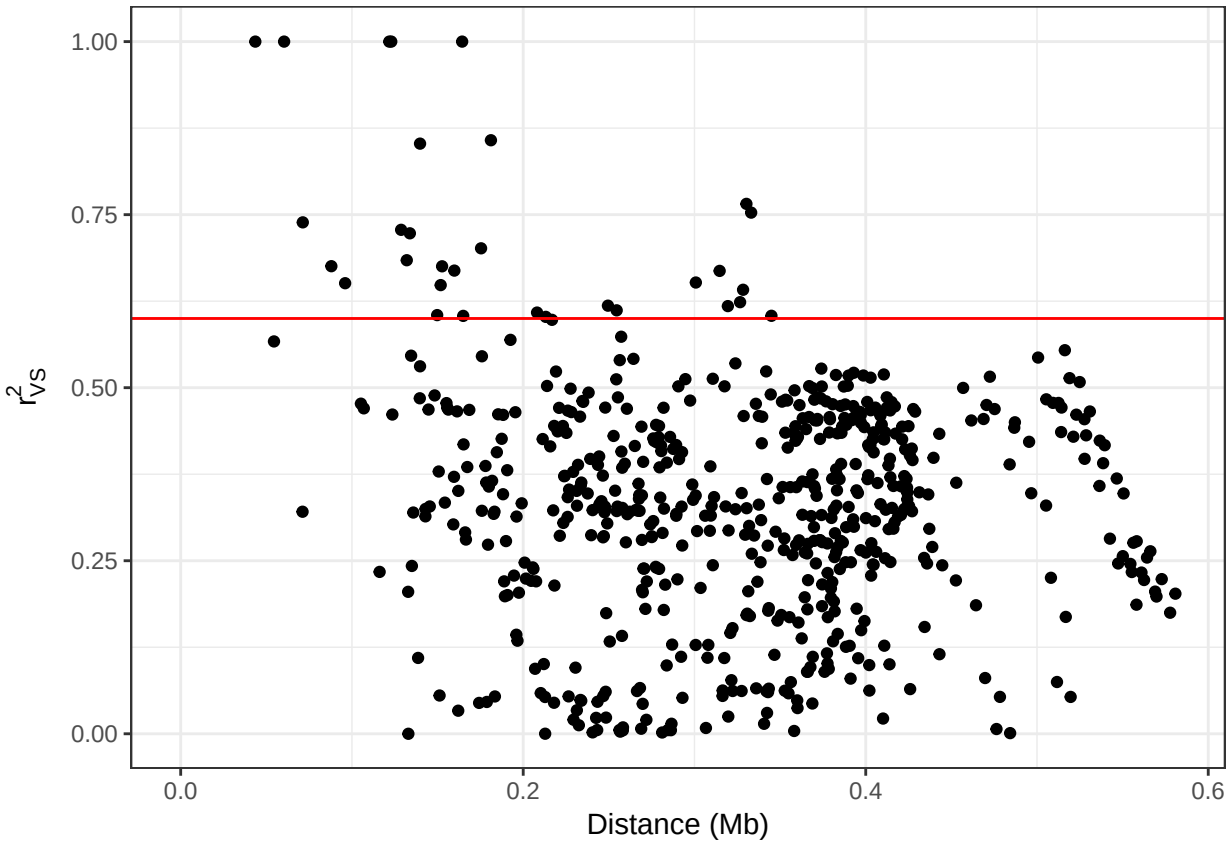


The same kind of graph shows that the number of markers has been reduced following clustering, removing redundant information:



Linkage disequilibrium and distance

The last type of graph that can be generated by `autohaplo` is a graph of the relationship between linkage disequilibrium and physical distance for a set of markers. These graphs can be useful to identify the patterns of LD in the region surrounding the gene of interest, and can thus help in choosing distance and LD thresholds for the analysis. Here is a graph showing these data for the set of markers following filtering in this analysis. In this graph, the red line shows the R^2 threshold that has been used in the analysis.



Hapmap files

The genotypes of the individuals for sets of markers generated throughout the analysis by `autohaplo` can also be output as hapmap files. For the purposes of this workshop, three hapmap files were output: one file with the genotypes for the final set of clustered markers, one with the genotypes for the set of markers represented by these clusters, and a hapmap file with the set of haplotypes that were generated and their "genotypes" at different positions.