Installation:

We distribute BSATOS as a convenient Dockerfile, which can be downloaded from: https://github.com/maypoleflyn/BSATOS/blob/master/Dockerfile.

Assuming to have docker correctly installed (more info here: https://docs.docker.com/) To install BSATOS, users just have to download the Docker file and save it in a folder of their liking. From that folder, the docker can be built, in Linux, with:

```
docker build -t bsatos .
```

similar commands are available for other operating systems.

Running BSATOS
The BSATOS container can be run using:

```
docker run -i -t bsatos /bin/bash
```

Once the container is started, BSATOS can be run with:

```
bsatos
```

which will give a list of possible commands and a command line guide. All the steps of the pipeline can be manually run one after the other, but we recommend to launch the full pipeline with:

```
bsatos all
```

which will create a bash script with all the commands of the pipeline that can then be invoked to run the whole analysis process.

We recommend to run the BSATOS container with the following command:

```
docker run --rm --user=$(id -u):$(id -g) -v local_folder:/BSATOS/data -i -t bsatos /bin/bash
```

where local_folder is a folder in the local machine that contains all the data. This folder would be mounted, within the BSATOS container, in /BSATOS/data and the owner of the data would be the local user.

BSATOS will create several files and folders during its execution. See below for a description of all the files.

Workflow description

In step I:

This step is performed using two commands: **'prepar'** and **'prep'**. The first is used to process parent data including SNVs and SV calling, variation annotation and SNVs classification. The second to obtain reads counts with different genotypes and conduct some filtering process.

Detailed information on each command:

The **'prepar'** command prepares the parents data as follows:

1) Align the reads from the two parents (pollen parent and maternal parent) to the genome reference using BWA (note that this is skipped if inputs are pre-aligned BAMs files);

2) Remove duplicates, index and sort BAMs files from 1) using SAMtools;
3) Perform SNP and InDel calling using SAMtools;
4) Perform SVs calling (bigger segment deletion and insertion) using Delly2;
5) Filter SNVs based on reads depth (default: min 10) and phred-scaled quality (default:30);
6) Filter SVs based support reads depth (default:10) and phred-scaled quality (default:30);
7) Annotate SNVs and SVs using ANNOVAR;
8) Split SNVs into three types (gP, gM and gMP, respectively present in the pollen, maternal sample and both) and obtain three types of SNVs sets.

The **'prep'** command prepares the pool data as follows:

1) Align the reads from two pools (High pool and Low pool) to the reference genome, using BWA (note that this step is skipped if BAM files are provided as input);
2) Remove duplicates, index and sort BAMs files from 1) using SAMtools;
3) Genotype BAMs of H and L pools using *SAMtools mpileup* respectively.
4) Filter SNVs based on the total read depth (default: min 10) and the phred-scaled quality (default: 30). For each pool, SNVs with support lower than 3 reads are considered noise and removed.
5) Extract read counts for H and L pools and merge them into three types of reads counts files (gP, gM and gMP)

In step II:

In the second module, haplotype blocks are assembled using paired-reads. The maximum-likelihood-based tool HapCUT2 or Hidden Markov Model-based algorithm integrated in SAMtools are used to assemble haplotype blocks from DNA sequence reads (Edge, et al., 2017; Li, 2011; Li, et al., 2009). This step is accomplished with one single command of the BSATOS pipeline:**'haplotype'**.

In a nutshell, HapCUT2 is a maximum-likelihood-based tool for assembling haplotypes from DNA sequence reads, designed to "just work" with excellent speed and accuracy. Besides NGS short reads, HapCUT2 support: clone-based sequencing (Fosmid or BAC clones), SMRT reads (PacBio),Oxford Nanopore reads,10X Genomics Linked-Reads,proximity-ligation (Hi-C) reads,
high-coverage sequencing (>40x coverage-per-SNP) using above technologies and combinations of the above technologies (e.g. scaffold long reads with Hi-C reads)

For NGS short reads data, in practice, SAMtools could construct longer haplotype blocks.

The **'haplotype'** command is used to construct and classify the haplotype blocks

If SAMtools is selected

1) Build four phased haplotype blocks (P.blocks, M.blocks, H.blocks and L.blocks) based on BAMs files from the two parents and H, L pools with *SAMtools phase* .
2) Filter SNVs located in haplotype blocks based on SNVs files from STEP1.
3) Sort four block files (P.blocks, M.blocks, H.blocks and L.blocks) and haplotype blocks keeping blocks with more reference alleles "on the left". Finally, the four block files are merged into one file.
4) Missing genotypes within haplotype blocks are imputed using a sliding window approach moving by one marker at a time and merged based on linkage relationships. A window harboring two adjacent makers slides one maker every time. Within one window, the missing genotype or gaps within the haplotype blocks were inferred and merged based on the linkage

relationship.
5) Compare and classify haplotype blocks of parents (Figure 1B).

If HapCUT2 is selected

1) Convert BAM files to the compact fragment file format containing only haplotype-relevant information with extractHAIRS. This is a necessary preparation step to running HapCUT2;
2) Assemble fragment files into haplotype blocks (P.blocks, M.blocks, H.blocks and L.blocks) with HAPCUT2;
3) Filter SNVs located in haplotype blocks based on SNVs files from step I.
4) Sort four block files (P.blocks, M.blocks, H.blocks and L.blocks) and haplotype blocks keeping blocks with more reference alleles "on the left". Finally, the four block files are merged into one file.
5) Missing genotypes within haplotype blocks are imputed using a sliding window approach moving by one marker at a time and merged based on linkage relationships. A window harboring two adjacent makers slides one maker every time. Within one window, the missing genotype or gaps within the haplotype blocks were inferred and merged based on the linkage relationship.
6) Compare and classify haplotype blocks of parents. .

Step III:

The three categories of markers (gP, gM and gMP) are processed in isolation. The commands used by BSATOS are **'afd', 'polish' and 'qtl_pick'**.

The **'afd'** command is used to calculate and filter the allele frequency difference between two extreme pools as follows:

1) The Allele frequency (AF) of each allele and G value are calculated for each SNV based on read counts.

2) The Nadaraya-Watson kernel regression is used as smoothing function to compute a G' value at each site within a sliding window having size defined by the user.

3) Non-parametric estimation of the null distribution of G'. A P value is assigned to each SNV. The false discovery race (FDR) of each SNVs is calculated and compared to a threshold (default: 0.01); Significant regions with FDR <0.01 are picked as candidate QTLs.

The **'polish'** command is used to polish candidate QTL regions and remove noisy markers based on haplotype information as follows:

[First of all, we define the reference allele frequency in each pool as RAF, the alternative allele frequency as AAF, the absolute value of the difference of the alternative allele frequency between two pools as AAFD. The sign of the difference of the alternative allele frequency of the two pools is AAFDS.]

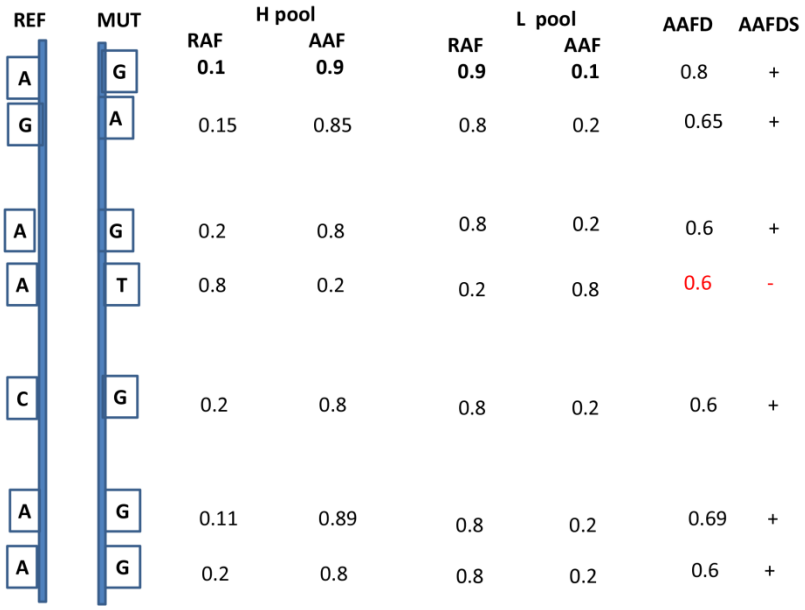


Figure S1
Scheme of the example of definition of AAF, AAFD and AAFDS

[For example, the first marker in Figure 1, we know that the reference allele is 'A' and the mutant allele is 'G', the AAF in H pool is 0.9 and the AAF in L pool is 0.1. so the AAFD=0.8 and the AAFDS should be '+';]

Markers of type gP, gM and gMP are processed in isolation as follows:

1. Based on the classified haplotype block information of the two parents (computed in step II) and allele frequency of each marker located in the haplotype blocks of a candidate QTL region, the haplotype blocks with at least 5 markers l and 70% of the markers showing consistent sign (AAFDS) are taken into account for further analysis. The other blocks are discarded.

Before processing further the kept blocks, markers showing inconsistent sign are removed.

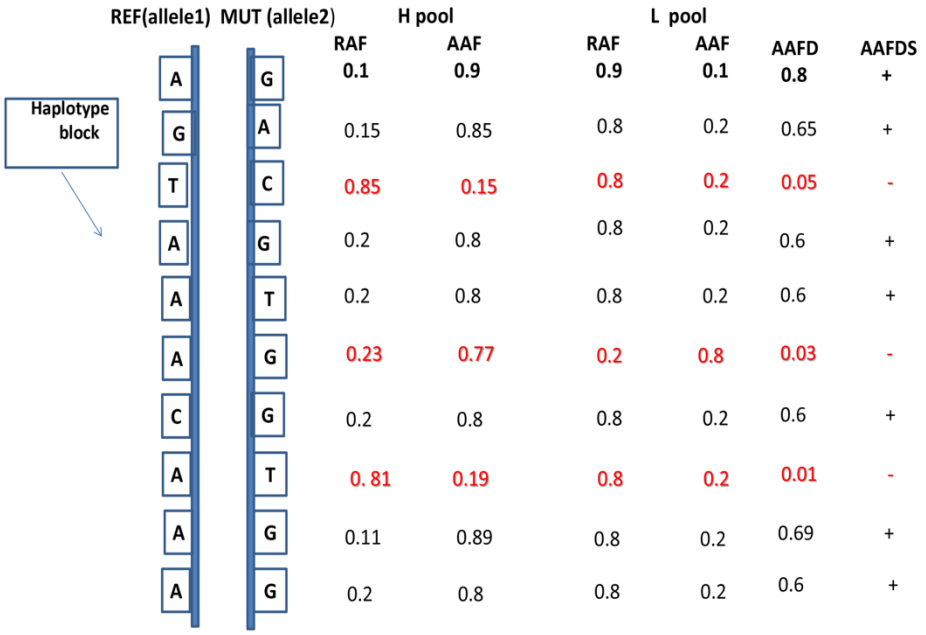


Figure S2
Scheme of the example of enriched haplotype block

In Figure S2, 10 markers are located in one haplotype block. 7 markers show consistent sign (AAFDS) are kept and the other three markers are discarded.

2. In the case of gMP markers (Figure S2A), if the phase of the marker linked with the functional mutation is not consistent between the two parents (Marker2 in Figure S2A), the marker is discarded as it does not produce observable differences between the two pools (both have an A:B allele ratio equal to 1:1).

3. The G statistic re-computed on all remaining markers is smoothed again with Nadaraya-Watson kernel regression using different window sizes (3w/4, w/2, w/4, w/8, w= user-defined window size).

The command **'qtl_pick'** is used to identify QTLs from the three types of peaks (P, M and MP), as follows:

1) Candidate QTL regions are identified as above.
2) QTL regions are refined using different sliding window sizes. For example, in the Figure S4, different sliding window sizes produce different G' profiles and the intersection of the signals is used to refine the QTL peak.
3) For each gene in proximity with the QTL region, the relative distance scores (RDS) is computed as follows

$$RDS = \quad \times 100$$

RDS: relative distance scores; ***pos***: the position of specific gene; ***peak***: the position of the peak; **w**: window size.

Genes with smaller RDS are more likely to be the functional genes underlying the interesting phenotype

4) The origins of the QTLs are identified by comparing the profiles (P, M and MP type) more present in the H/L pool. If a QTL is detected by more than 2 profiles, the one with higher G' is regarded as the origin of the QTL.

5) Based on the origin of QTLs and RDS, candidate genes and the functional mutations underlying QTLs are reported.

The command **'igv'** is used to generate files for Integrative Genomics Viewer (IGV)

In order to help users to explore the results of BASTOS, it produces an output that can be directly imported into IGV (see below).

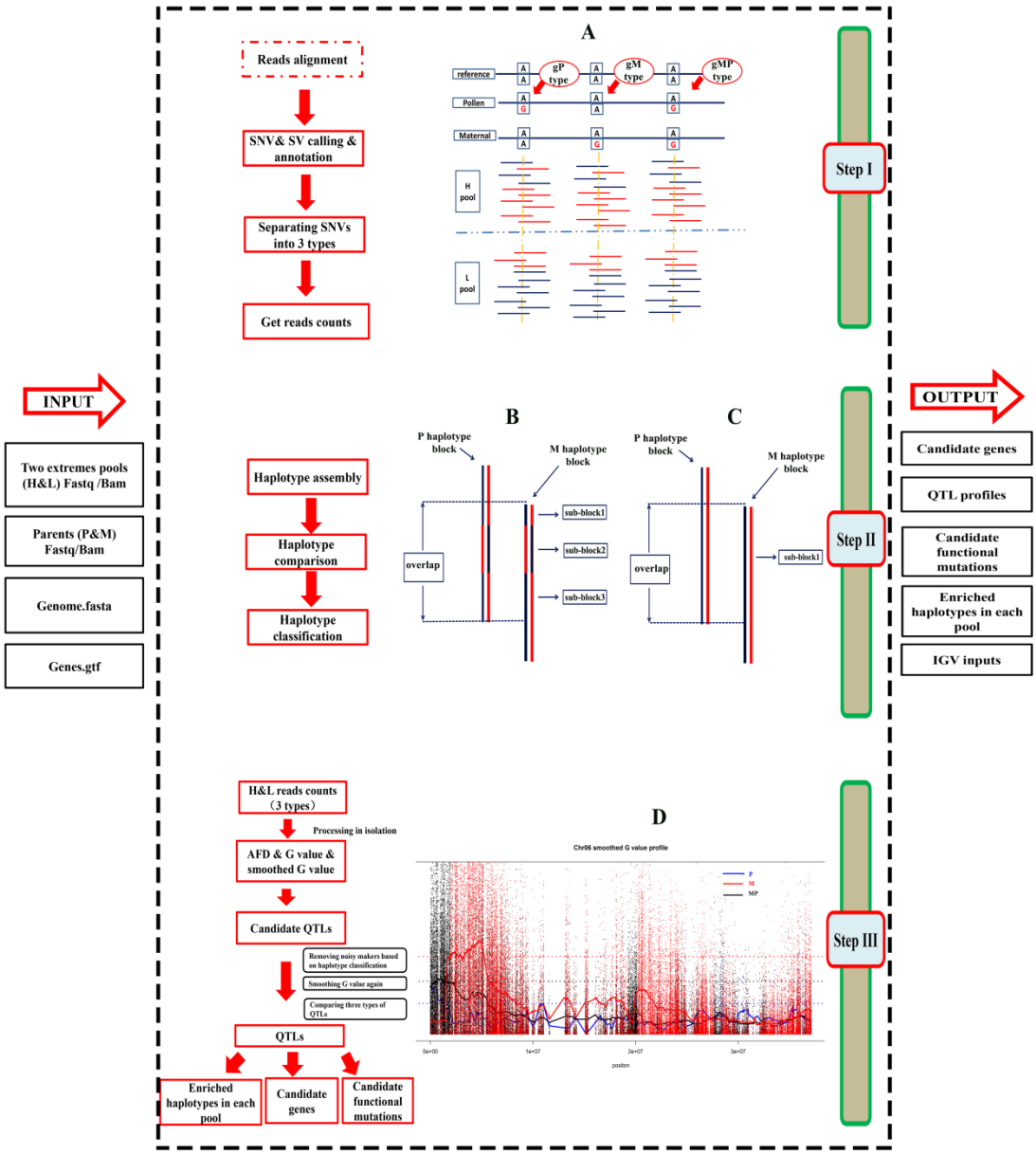


Figure S3

Scheme of BSATOS

A: the schema of three types of makers and getting reads counts harboring different genotypes
B, C: the schema of haplotype classification
D: the QTLs profiles of example data

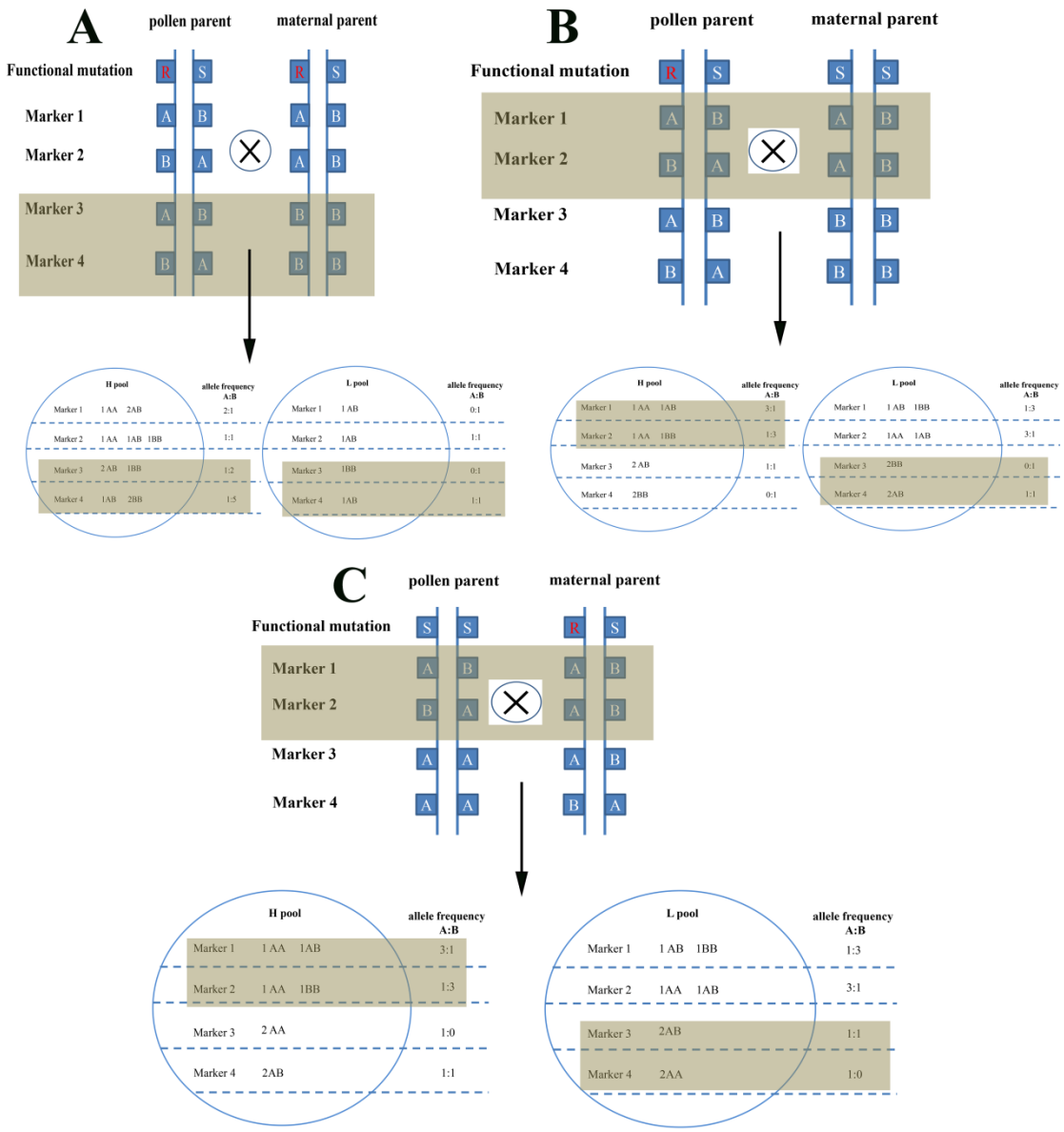


Figure S4

Scheme of segregation rule of different types of markers in different circumstances

A) The functional mutation underlying QTLs is double heterozygous (PM type) and the origin of QTL is from both parents. B) and C) the functional mutations underlying QTLs is single heterozygous (P or M type) and the origin of QTL is either pollen parent or maternal parent. Maker1 – Maker4 represent four different type of SNPs, A for one type of base (A,T,G,C) and B for other one (A,T,G,C)

Here, we analyze the segregation rule of different types of markers (P, M and PM) in different circumstances.

A) When the functional mutation underlying QTLs (R) is double heterozygous (MP type) and the origin of QTL is from both parents, there are two double heterozygous markers (Marker1 and Marker2) and two P type markers in the figure. The phase between Marker1 and R are consistent with parents, however, the phase between Marker2 and R are inconsistent with parents. When the R locus is complete dominant to S locus, after extreme progenies selection, the allele frequency of R in H pool should be 3/4 and the allele frequency of R in L pool should be 1/4. When Marker1 is close linked with R locus, the allele frequency of allele A in H pool should be 3/4, the allele frequency of allele A in L pool should be 1/4 and allele frequency difference of allele A should be 1/2. However, when Marker2 is also close linked with R locus, the allele frequency of allele A both in H and L pools are 1/2 and the allele frequency difference is 0. For other two P type markers (Marker3 and Marker4), When Marker3 is close linked with R locus, the allele frequency of allele A in H pool should be 1/3, the allele frequency of allele A in L pool should be 0 and the allele frequency difference should be 1/3.

B), C) the functional mutations underlying QTLs (R) is heterozygous (P or M type) in one of the parents and the origin of QTL is either pollen parent or maternal parent. When the R locus is complete dominant to S locus, after extreme progenies selection, when Markers are close linked with R locus, the allele frequency difference of allele A between H and L pools should be 1/2.

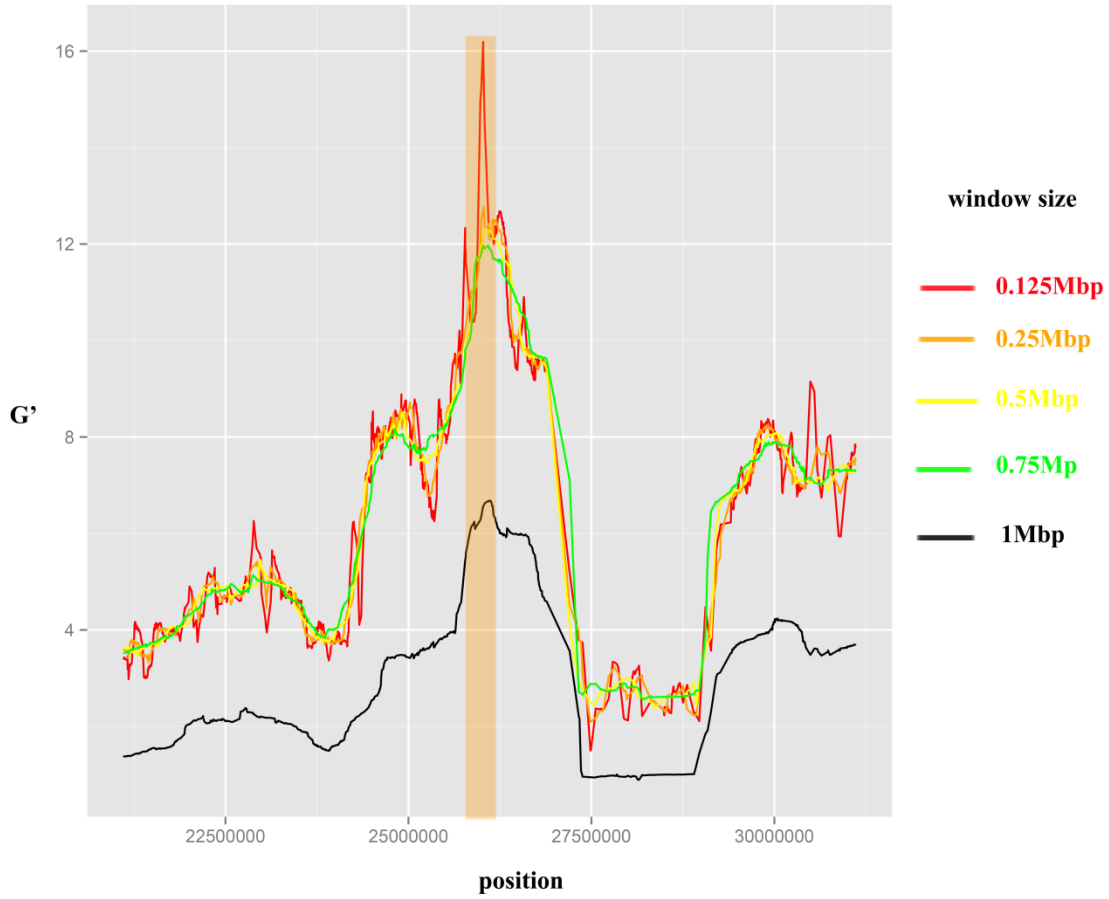


Figure S5

The profile of detect QTL regions using different sliding window sizes.

DESCRIPTION OF OUTPUT:

The **'prepar'** command prepares the parents data.

Outputs

By default, all the result will be kept in the **'prepar_dir'.**

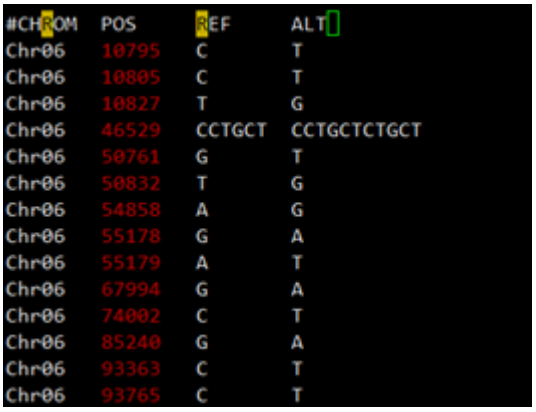
`Data structure:`

prepar_dir[DIR]

```
        |
        |--summary [FILE]
        |--P_M.snv        [FILE]
         |--sv.vcf  [FILE]
         |--P_M.m   [FILE]
         |--P_M.p   [FILE]
         |--P_M.pm  [FILE]
         |--anno [DIR]   (annotated files are all included in this directory)
                  |
```

|--snv.AT_multianno.txt [FILE] (annotated SNVs file)
|--snv.AT_multianno.vcf [FILE] (annotated SNVs VCF file)
|--snv.avinput [FILE] (inputs of SNVs annotation)
|--sv.AT_multianno.txt [FILE] (annotated SVs file)
|--sv.avinput [FILE] (inputs of SVs annotation)
| -- sv.AT_multianno.vcf (annotated SVs VCF file)
|--AT_refGeneMrna.fa [FILE] (annotation database)
|--AT_refGene.txt [FILE] (annotation database)

P_M.p [FILE]
The genotype of the markers are homozygous in maternal parent but are heterozygous in pollen parent
The first part of the file::



#CHROM	POS	REF	ALT
Chr06	10795	C	T
Chr06	10805	C	T
Chr06	10827	T	G
Chr06	46529	CCTGCT	CCTGCTCTGCT
Chr06	50761	G	T
Chr06	50832	T	G
Chr06	54858	A	G
Chr06	55178	G	A
Chr06	55179	A	T
Chr06	67994	G	A
Chr06	74002	C	T
Chr06	85240	G	A
Chr06	93363	C	T
Chr06	93765	C	T

From left to right by column:

CHROM: the chromosome of this SNV
POS: the position of the SNVs in the chromosome
REF: the allele of reference
ALT: the allele of alter.

P_M.m [FILE]
The genotype of the markers are homozygous in pollen parent but is heterozygous in maternal parent
The first part of the file::

#CHROM	POS	REF	ALT
Chr06	42	C	T
Chr06	56	G	T
Chr06	82	A	C
Chr06	101	C	T
Chr06	156	A	G
Chr06	160	G	A
Chr06	221	C	T
Chr06	240	G	A
Chr06	261	C	T
Chr06	283	G	T
Chr06	287	C	G
Chr06	319	A	G

From left to right by column:

CHROM: the chromosome of this SNV
POS: the position of the SNVs in the chromosome
REF: the allele of reference
ALT: the allele of alter.

P_M.pm [FILE]
The genotypes of the markers are both heterozygous

#CHROM	POS	REF	ALT
Chr06	133	G	A
Chr06	141	C	T
Chr06	522	G	A
Chr06	530	A	G
Chr06	1282	T	C
Chr06	2703	A	G
Chr06	3044	T	C
Chr06	4733	T	C
Chr06	9567	T	G
Chr06	12828	C	T
Chr06	13011	G	A

From left to right by column:
CHROM: the chromosome of this SNV
POS: the position of the SNVs in the chromosome
REF: the allele of reference
ALT: the allele of alter.

M_P.snv [FILE]
The SNVs VCF file of two parents

sv.vcf [FILE]
The SVs VCF file of two parents

anno [DIR]
Annotated files are all included in this directory

AT_refGene.txt [FILE]
GenePred file

AT_refGeneMrna.fa [FILE]
transcript FASTA file

snv.AT_multianno.txt [FILE]
SNVs multianno file

snv.AT_multianno.vcf [FILE]
SNVs annotated VCF file

snv.avinput [FILE]
SNVs input file

sv.AT_multianno.txt [FILE]
SVs multianno file

sv.AT_multianno.vcf [FILE]
SVs annotated VCF file

sv.avinput [FILE]
SVs input file

summary [FILE]
summary of the reads alignment of each parents
The number of SNVs &SVs
The number of SNVs in P, M and PM type.

The **'prep'** command prepares the pool data

```
prep_dir[DIR]
        |
          |
          |--M.counts  [FILE] read counts with different alleles from H & L pools in M type loci
          |
          |--P.counts  [FILE] read counts with different alleles from H & L pools in P type loci
          |
          |--PM.counts [FILE] read counts with different alleles from H & L pools in PM type loci
          |
          |--sum [FILE] summary data of M.counts, P.counts and PM.counts
```

*.counts [FILE]

read counts file with different alleles from H & L pools in P, M and PM type loci

The first part of the files is as followings:

From left to right by column:
Chromosome: the chromosome of this SNV
Position the position of the SNVs in the chromosome
H_REF: reads counts with reference alleles in H pool
H_ALT: reads counts with alter alleles in H pool
L_REF: reads counts with reference alleles in L pool
L_ALT: reads counts with alter alleles in L pool

The command **'haplotype'** is used to construct haplotype blocks.

```
haplotype_dir[DIR]
        |
          |--M_block          [FILE]  haplotype blocks of maternal parent
          |--P_block          [FILE]  haplotype blocks of pollen parent
          |--M_haplotype.bed  [FILE]  BED format haplotype information of maternal parent
          |--P_haplotype.bed  [FILE]  BED format haplotype information of pollen parent
          |--overlapped.bed   [FILE]  BED format haplotype information classified from two parent
          |--sub_haplotype    [FILE]  haplotype sub-blocks information within haplotype block
          |--haplotype.block  [FILE]  merged, corrected and patched haplotype blocks
```

*_block [FILE]
haplotype blocks of maternal parent, pollen parent

The first part of the file is as followings:

From left to right by column:
CHROM: the chromosome of this SNV
POS: the position of the SNVs in the chromosome
REF: the allele of reference
ALT: the allele of alter
HAP2: allele in haplotype1 [0 for reference allele; 1 for alter allele]
HAP1: allele in haplotype2 [0 for reference allele; 1 for alter allele]
NAME: haplotype name

*_haplotype.bed [FILE]

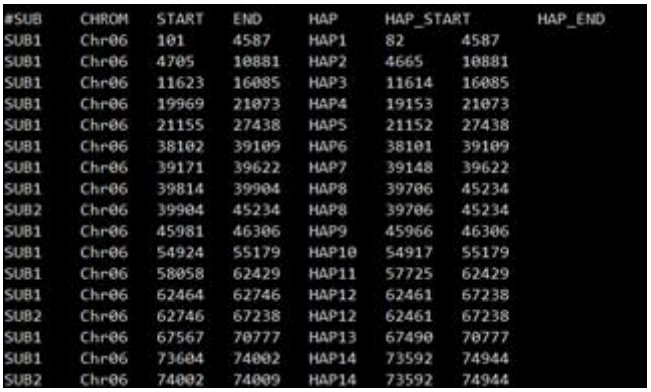
BED format haplotype information of maternal parent

overlapped.bed [FILE]

BED format haplotype information classified from two parents and two pools.

sub_haplotype [FILE]
haplotype sub-blocks information within haplotype block

The first part of the file:

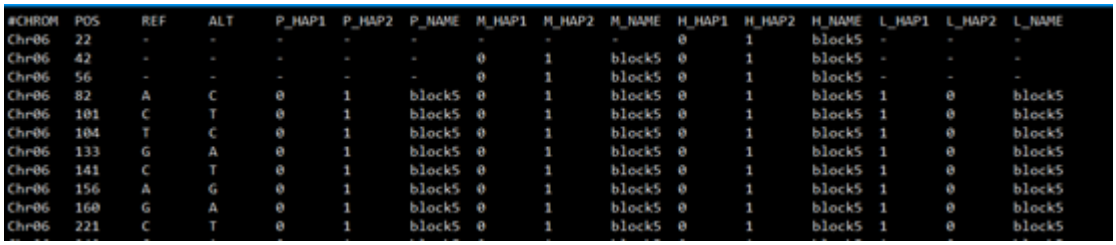


#SUB	CHROM	START	END	HAP	HAP_START	HAP_END
SUB1	Chr06	101	4587	HAP1	82	4587
SUB1	Chr06	4705	10881	HAP2	4665	10881
SUB1	Chr06	11623	16085	HAP3	11614	16085
SUB1	Chr06	19969	21073	HAP4	19153	21073
SUB1	Chr06	21155	27438	HAP5	21152	27438
SUB1	Chr06	38102	39109	HAP6	38101	39109
SUB1	Chr06	39171	39622	HAP7	39148	39622
SUB1	Chr06	39814	39904	HAP8	39706	45234
SUB2	Chr06	39904	45234	HAP8	39706	45234
SUB1	Chr06	45981	46306	HAP9	45966	46306
SUB1	Chr06	54924	55179	HAP10	54917	55179
SUB1	Chr06	58058	62429	HAP11	57725	62429
SUB1	Chr06	62464	62746	HAP12	62461	67238
SUB2	Chr06	62746	67238	HAP12	62461	67238
SUB1	Chr06	67567	70777	HAP13	67490	70777
SUB1	Chr06	73604	74002	HAP14	73592	74944
SUB2	Chr06	74002	74009	HAP14	73592	74944

From left to right by column:

SUB: sub-blocks name in the haplotype block
CHROM: the chromosome of sub-block
START: the start position of sub-block in chromosome
END: the end position of sub-block in chromosome
HAP: haplotype block name
HAP_START: the start position of haplotype block
HAP_END: the end position of haplotype block

haplotype.block [FILE]



#CHROM	POS	REF	ALT	P_HAP1	P_HAP2	P_NAME	M_HAP1	M_HAP2	M_NAME	H_HAP1	H_HAP2	H_NAME	L_HAP1	L_HAP2	L_NAME
Chr06	22	-	-	-	-	-	-	-	-	0	1	block5	-	-	-
Chr06	42	-	-	-	-	-	0	1	block5	0	1	block5	-	-	-
Chr06	56	-	-	-	-	-	0	1	block5	0	1	block5	-	-	-
Chr06	82	A	C	0	1	block5	0	1	block5	0	1	block5	1	0	block5
Chr06	101	C	T	0	1	block5	0	1	block5	0	1	block5	1	0	block5
Chr06	104	T	C	0	1	block5	0	1	block5	0	1	block5	1	0	block5
Chr06	133	G	A	0	1	block5	0	1	block5	0	1	block5	1	0	block5
Chr06	141	C	T	0	1	block5	0	1	block5	0	1	block5	1	0	block5
Chr06	156	A	G	0	1	block5	0	1	block5	0	1	block5	1	0	block5
Chr06	160	G	A	0	1	block5	0	1	block5	0	1	block5	1	0	block5
Chr06	221	C	T	0	1	block5	0	1	block5	0	1	block5	1	0	block5

From left to right by column:

CHROM: The chromosome of the SNV
POS: The position of the SNV in chromosome
REF: The reference allele
ALT: the alter allele
P_HAP1: allele in haplotype1 [0 for reference allele; 1 for alter allele] of pollen parent
P_HAP2: allele in haplotype2 [0 for reference allele; 1 for alter allele] of pollen parent
P_NAME: haplotype name in pollen parent
M_HAP1: allele in haplotype1 [0 for reference allele; 1 for alter allele] of maternal parent
M_HAP2: allele in haplotype1 [0 for reference allele; 1 for alter allele] of maternal parent
M_NAME: haplotype name in maternal parent
H_HAP1: allele in haplotype1 [0 for reference allele; 1 for alter allele] of H pool
H_HAP2: allele in haplotype1 [0 for reference allele; 1 for alter allele] of H pool
H_NAME: haplotype name in H pool
L_HAP1: allele in haplotype1 [0 for reference allele; 1 for alter allele] of L pool
L_HAP2: allele in haplotype1 [0 for reference allele; 1 for alter allele] of L pool
L_NAME: haplotype name in L pool

The **'afd'** command is used to calculate and filter allele frequency difference between two extreme pools

AFD_dir[DIR]
|
|--P.AFD [FILE]
|--M.AFD [FILE]
|--PM.AFD [FILE]

P.AFD [FILE]
G value based P type loci and smoothed curve with different window across genome with haplotype information

M.AFD [FILE]
G value based M type loci and smoothed curve with different window across genome with haplotype information

PM.AFD [FILE]
G value based PM type loci and smoothed curve with different window across genome with haplotype information

The first part of the *_AFD:

```
Column1-10:
```

#CHROM	POS	H_REF	H_ALT	L_REF	L_ALT	H_REF_AF	H_ALT_AF	L_REF_AF	L_ALT_AF
Chr06	67994	33	2	19	11	0.942857142857143	0.0571428571428571	0.633333333333	
Chr06	85240	27	10	37	12	0.72972972972973	0.27027027027027	0.755102040816	
Chr06	93951	37	10	50	11	0.787234042553192	0.212765957446809	0.819672131147	
Chr06	93975	37	8	49	16	0.822222222222222	0.177777777777778	0.753846153846	
Chr06	94247	57	17	72	32	0.77027027027027	0.22972972972973	0.692307692307	
Chr06	94475	73	23	71	26	0.760416666666667	0.239583333333333	0.731958762886	
Chr06	94618	30	17	26	17	0.638297872340426	0.361702127659574	0.604651162790	
Chr06	94662	26	8	24	13	0.764705882352941	0.235294117647059	0.648648648648	
Chr06	94684	26	10	21	11	0.722222222222222	0.277777777777778	0.65625 0.3437	
Chr06	103937	19	12	19	16	0.612903225806452	0.387096774193548	0.542857142857	

Columnll-l5：

G_VALUE	Gprimer_1M	Gprimer_0.75M	Gprimer_0.5M	Gprimer_0.25M
3333	0.366666666666667	10.2905728448925		1.12078438298491
6326	0.244897959183673	0.0710953419847191		1.1639169608415
7541	0.180327868852459	0.177527978592913		1.1642558145198
6154	0.246153846153846	0.741611025329638		1.1642605953467
7692	0.307692307692308	1.3341306967514	1.16431474529776	
6598	0.268041237113402	0.206385690465245		1.1643600893114
0698	0.395348837209302	0.108121385489232		1.1643885071662
8649	0.351351351351351	1.15521019779262		1.1643972477723
75	0.345129833744719	1.16440161748435	0.995764430851408	
7143	0.457142857142857	0.330815379367888		1.1662044931636
1.03419227909791		1.16621270820538	0.995287040402043	
3529	0.382352941176471	1.17351428570143		1.1662209217354
3333	0.366666666666667	1.35564661685504		1.1662235955818
4615	0.384615384615385	2.04241564610407		1.1662289427939

Column16—
haplotype.block information

From left to right by column:

CHROM：The chromosome of the marker
POS：The position of the marker in the chromosome
H_REF: read counts with reference allele in H pool
H_ALT: read counts with alter allele in H pool
L_REF; read counts with reference allele in L pool
L_ALT: read counts with alter allele in L pool
H_REF_AF: allele frequency of reference allele in H pool
H_REF_ALT: allele frequency of alter allele in H pool
L_REF_AF: allele frequency of reference allele in L pool
L_REF_ALT: allele frequency of alter allele in L pool
G_VALUE: G value
Gprimer_1M: G' value with the one window size (default: 1Mbp)
Gprimer_0.75M: G' value with 3/4 window size (default: 0.75Mbp)
Gprimer_0.5M: G' value with 1/2 window size (default: 0.5Mbp)
Gprimer_0.25M: G' value with 1/4 window size (default: 0.25Mbp)

The command **'polish'** is used to polish candidate QTLs regions and remove noisy makers based on haplotype information

polish_dir[DIR]
|
|
|--M.polished.afd [FILE]
|--P.polished.afd [FILE]
|--PM.polished.afd [FILE]
|--m.igv [FILE]

|--p.igv [FILE]
|--pm.igv [FILE]

P.polished.afd [FILE]
The format is the same as P.AFD
G value based on P type loci (after removing noisy) and smoothed curve with different window across genome with haplotype information

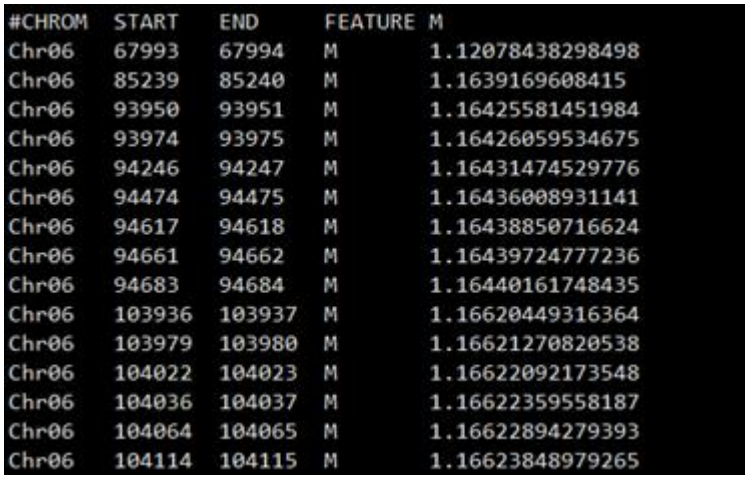
M.polished.afd [FILE]
The format is the same as M.AFD
G value based on M type loci (after removing noisy) and smoothed curve with different window across genome with haplotype information

PM.polished.afd [FILE]
The format is the same as PM.AFD
G value based on PM type loci (after removing noisy) and smoothed curve with different window across genome with haplotype information

m.igv [FILE]

G' value profiles of M type loci could be visualized by Integrative Genomics Viewer (IGV)

The first part is as followings:



#CHROM	START	END	FEATURE	M
Chr06	67993	67994	M	1.12078438298498
Chr06	85239	85240	M	1.1639169608415
Chr06	93950	93951	M	1.16425581451984
Chr06	93974	93975	M	1.16426059534675
Chr06	94246	94247	M	1.16431474529776
Chr06	94474	94475	M	1.16436008931141
Chr06	94617	94618	M	1.16438850716624
Chr06	94661	94662	M	1.16439724777236
Chr06	94683	94684	M	1.16440161748435
Chr06	103936	103937	M	1.16620449316364
Chr06	103979	103980	M	1.16621270820538
Chr06	104022	104023	M	1.16622092173548
Chr06	104036	104037	M	1.16622359558187
Chr06	104064	104065	M	1.16622894279393
Chr06	104114	104115	M	1.16623848979265

From left to right by column:

CHROM: The chromosome of the marker
START: The start position of the marker
END: The end position of the marker
FEATURE: The marker type [P or M or PM]
M: The G' value of this marker

p.igv [FILE]

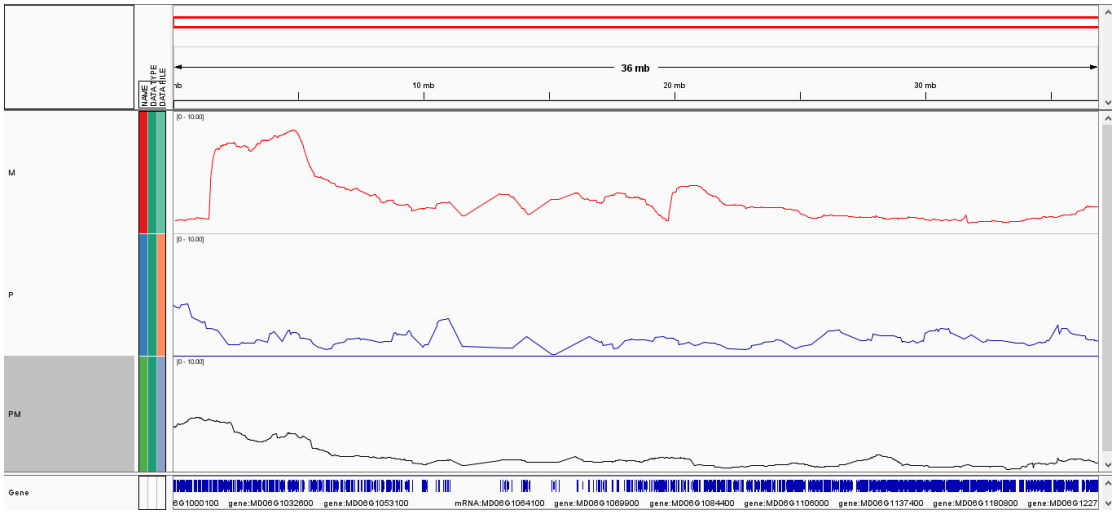
The format is the same as m.igv

G' value profiles of P type loci could be visualized by Integrative Genomics Viewer (IGV)

pm.igv [FILE]

The format is the same as m.igv

G’ value profiles of PM type loci could be visualized by Integrative Genomics Viewer (IGV)



Screenshot of the files loaded with IGV

The command **‘qtl_pick’** is used to judge and pick up QTLs from three types of peaks.

```
qtl_pick_dir [DIR]
                |
                |
                |--qtl [FILE]
                |--*pdf [FILE]
                |--P_hap [FILE]
                |--M_hap [FILE]
                |--PM_hap [FILE]
                |--gene.bed [FILE]
                |*.gene [FILE]
                |*.hap[FILE]
                |*.snv [FILE]
                |*.snv.igv.mut  [FILE]
                |*.snv.igv.vcf   [FILE]
                |*.sv [FILE]
                |*.sv.igv.mut   [FILE]
                |*.sv.igv.vcf  [FILE]
```

qtl	[FILE]	detected QTLs list file
*.pdf	[FILE]	G' value profiles across each chromosome (*:chromosome)
*.pdf	[FILE]	multiple G' values profiles across QTL region (*:QTL accession)
p_hap	[FILE]	enriched haplotype information in P type loci
m_hap	[FILE]	enriched haplotype information in M type loci
pm_hap	[FILE]	enriched haplotype information in PM type loci
*.hap	[FILE]	haplotype information in each QTL region (*: QTL accession)
*.gene	[FILE]	gene list located in the QTL regions (*: QTL accession)
*.snv	[FILE]	screened SNVs based on genetic rules located in the QTL regions (*:QTL accession)
*.snv.igv.mut	[FILE]	screened SNVs based on genetic rules located in the QTL regions (*:QTL accession) [MUT format]
*.snv.igv.vcf	[FILE]	screened SNVs based on genetic rules located in the QTL regions (*:QTL accession) [VCF format]
*.sv	[FILE]	screened SNVs based on genetic rules located in the QTL regions (*:QTL accession)
*.sv.igv.mut	[FILE]	screened SVs based on genetic rules located in the QTL regions (*:QTL accession) [MUT format]
*.sv.igv.vcf	[FILE]	screened SVs based on genetic rules located in the QTL regions (*:QTL accession) [VCF format]

.thres [FILE] threshold information (: p/m/pm)

```
7.303028
7.341369
7.371755
7.43597
```

From top to bottom:

The threshold value from G' value smoothed with one sliding window size (user-defined)
The threshold value from G' value smoothed with 3/4 sliding window size
The threshold value from G' value smoothed with 1/2 sliding window size
The threshold value from G' value smoothed with 1/4 sliding window size

qtl [FILE]

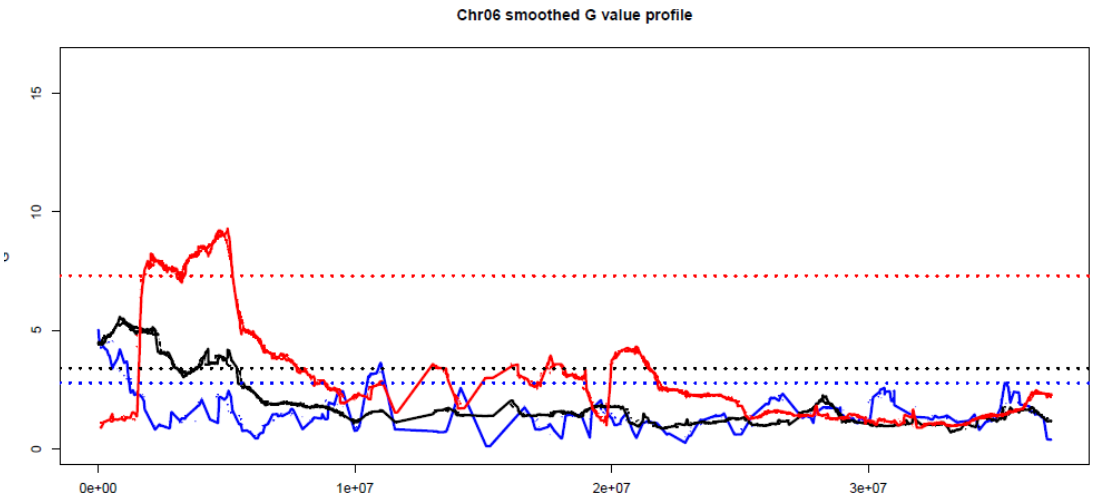
```
#Origin CHROM   QTL_START       QTL_END PEAK_POS        PEAK    ACCESSION
P       Chr06   10027489        10527489        10527489        3.059511        P1
P       Chr06   11009860        11509860        11009860        3.627277        P2
M       Chr06   1812818 2563869 2063869 8.240285        M1
M       Chr06   4542571 5235914 5042571 9.296073        M2
H       Chr06   3058865 3558865 3058865 3.423959        H1
```

From left to right by column:

Origin: the origin of QTL
CHROM: the chromosome of the QTL
QTL_START: the start position of the QTL

QTL_END: the end position of the QTL
PEAK_POS: the peak position of the QTL
PEAK: the G' value in the peak position
ACCESSION: the QTL accession

*.pdf [FILE]
G value profiles across each chromosome (*:chromosome)



The example G value profiles across chromosome 6

*.pdf [FILE]
multiple G' values profiles across QTL region (*.QTL accession)

See Figure S5

p_hap [FILE]
haplotype information in P type loci

#CHROM	POS	REF	ALT	P_HAP1	P_HAP2	P_NAME	P_EN	P_G	M_HAP1	M_HAP2	M_NAME	M_EN	M_G				
Chr06	15115882		A	C	0	1	block7314		H	7.18019122824675		1	0	block7316		L	7.18
Chr06	15116384		A	G	0	1	block7314		H	7.18019122824675		1	0	block7316		L	7.18
Chr06	15116392		T	A	0	1	block7314		H	7.18019122824675		1	0	block7316		L	7.18
Chr06	15116410		T	G	0	1	block7314		H	7.18019122824675		1	0	block7316		L	7.18
Chr06	15116452		C	T	1	0	block7314		H	7.18019122824675		0	1	block7316		L	7.18
Chr06	15116453		T	C	0	1	block7314		H	7.18019122824675		1	0	block7316		L	7.18
Chr06	15116482		A	T	0	1	block7314		H	7.18019122824675		1	0	block7316		L	7.18
Chr06	15116495		A	G	0	1	block7314		H	7.18019122824675		1	0	block7316		L	7.18
Chr06	15116503		C	T	0	1	block7314		H	7.18019122824675		1	0	block7316		L	7.18
Chr06	15116505		G	T	0	1	block7314		H	7.18019122824675		1	0	block7316		L	7.18
Chr06	15116565		A	T	0	1	block7314		H	7.18019122824675		1	0	block7316		L	7.18
Chr06	15116566		A	G	0	1	block7314		H	7.18019122824675		1	0	block7316		L	7.18
Chr06	15116581		T	A	0	1	block7314		H	7.18019122824675		1	0	block7316		L	7.18

From left to right by column:

CHROM: The chromosome of the marker
POS: The position of the SNV in chromosome
REF: The reference allele
ALT: The alter allele
P_HAP1: allele in haplotype1 [0 for reference allele; 1 for alter allele] of pollen parent

P_HAP2: allele in haplotype2 [0 for reference allele; 1 for alter allele] of pollen parent
P_NAME: haplotype name in pollen parent
P_EN: the P_HAP2 enriched in H/L pool
P_G: averaged G' value in the P haplotype
M_HAP1: allele in haplotype1 [0 for reference allele; 1 for alter allele] of maternal parent
M_HAP2: allele in haplotype1 [0 for reference allele; 1 for alter allele] of maternal parent
M_NAME: haplotype name in maternal parent
M_EN: the M_HAP2 enriched in H/L pool
M_G: averaged G' value in the M haplotype

m_hap [FILE]
haplotype information in M type loci

The format is the same as p_hap.

pm_hap [FILE]

haplotype information in MP type loci

The format is the same as p_hap.

*.gene [FILE]
gene list located in the QTL regions (*: QTL accession)

#CHROM	START	END	GENE	PEAK	RDS
Chr06	4582147	4584370	MD06G1035500	5042571	92.0848
Chr06	4620597	4622648	MD06G1035600	5042571	84.3948
Chr06	4633833	4636732	MD06G1035700	5042571	81.7476
Chr06	4639129	4641015	MD06G1035800	5042571	80.6884
Chr06	4642129	4644365	MD06G1035900	5042571	80.0884
Chr06	4651894	4655813	MD06G1036000	5042571	78.1354
Chr06	4656091	4659669	MD06G1036200	5042571	77.296
Chr06	4661099	4662310	MD06G1036300	5042571	76.2944
Chr06	4663160	4667876	MD06G1036400	5042571	75.8822
Chr06	4675577	4679818	MD06G1036600	5042571	73.3988
Chr06	4683129	4683413	MD06G1036700	5042571	71.8884
Chr06	4690441	4696360	MD06G1036800	5042571	70.426
Chr06	4699553	4701512	MD06G1036900	5042571	68.6036
Chr06	4705225	4708485	MD06G1037000	5042571	67.4692
Chr06	4710491	4713008	MD06G1037100	5042571	66.416
Chr06	4735916	4736995	MD06G1037200	5042571	61.331
Chr06	4747254	4748441	MD06G1037300	5042571	59.0634

From left to right by column:

CHROM: The chromosome of the gene located in
START: The start position of the gene located in the chromosome
END: The end position of the gene located in the chromosome
GENE: The gene name
PEAK: The peak position of the profile peak
RDS: The RDS score. The smaller the better. (0-100)

*.snv [FILE]
screened SNVs based on genetic rules located in the QTL regions (*:QTL accession)

Column1-11:

#Chr	Start	End	Ref	Alt	Func.refGene	Gene.refGene	GeneDetail.refGene	ExonicFunc.refGene	AAChange.refGene			Othe
Chr06	3068474	3068478	TTTTT	-	intergenic	gene:MD06G1025200;gene:MD06G1025300	dist=13067;dist=1457	.	.	0.5		30.4
Chr06	3068476	3068480	TTTGT	-	intergenic	gene:MD06G1025200;gene:MD06G1025300	dist=13069;dist=1455	.	.	0.5		31.5
Chr06	3068598	3068598	C	T	intergenic	gene:MD06G1025200;gene:MD06G1025300	dist=13191;dist=1337	.	.	0.5		477
Chr06	3070088	3070097	GAAAAAGAAA	-	UTR3	gene:MD06G1025300	mRNA:MD06G1025300:c.*169_*160delTTTCTTTTTC	.	.	0.5		
Chr06	3070307	3070307	G	C	exonic	gene:MD06G1025300	.	nonsynonymous SNV	gene:MD06G1025300:mRNA:MD06G1025300:exon3:c.			
Chr06	3070840	3070845	AGGAGG	-	exonic	gene:MD06G1025300	.	nonframeshift deletion	gene:MD06G1025300:mRNA:MD06G1025300:exon1:c.			
Chr06	3071122	3071122	G	A	UTR5	gene:MD06G1025300	mRNA:MD06G1025300:c.-147C>T	.	.	0.5	476	53
Chr06	3071248	3071248	C	T	upstream	gene:MD06G1025300	dist=103	.	.	0.5	477	63
Chr06	3071654	3071654	T	G	upstream	gene:MD06G1025300	dist=509	.	.	0.5	477	58
Chr06	3071885	3071885	C	T	upstream	gene:MD06G1025300	dist=740	.	.	0.5	477	65
Chr06	3071894	3071894	G	A	upstream	gene:MD06G1025300	dist=749	.	.	0.5	477	57
Chr06	3071940	3071940	C	T	upstream	gene:MD06G1025300	dist=795	.	.	0.5	359	33
Chr06	3072058	3072058	A	C	upstream	gene:MD06G1025300	dist=913	.	.	0.5	477	57
Chr06	3072311	3072311	-	ATA	intergenic	gene:MD06G1025300;gene:MD06G1025400	dist=1166;dist=6411	.	.	0.5		217
Chr06	3072371	3072371	A	G	intergenic	gene:MD06G1025300;gene:MD06G1025400	dist=1226;dist=6351	.	.	0.5		477
Chr06	3072376	3072376	A	G	intergenic	gene:MD06G1025300;gene:MD06G1025400	dist=1231;dist=6346	.	.	0.5		477
Chr06	3072427	3072427	G	T	intergenic	gene:MD06G1025300;gene:MD06G1025400	dist=1282;dist=6295	.	.	0.5		477
Chr06	3077601	3077601	G	A	intergenic	gene:MD06G1025300;gene:MD06G1025400	dist=6456;dist=1121	.	.	0.5		477
Chr06	3078145	3078145	C	T	upstream	gene:MD06G1025400	dist=577	.	.	0.5	477	82
Chr06	3080370	3080370	T	G	exonic	gene:MD06G1025400	.	nonsynonymous SNV	gene:MD06G1025400:mRNA:MD06G1025400:exon2:c.			
Chr06	3080980	3080980	C	T	UTR3	gene:MD06G1025400	mRNA:MD06G1025400:c.*560C>T	.	.	0.5	477	57
Chr06	3080997	3080997	T	G	UTR3	gene:MD06G1025400	mRNA:MD06G1025400:c.*577T>G	.	.	0.5	477	65

Column12-25：

#CHROM	POS	REF	ALT	P_HAP1	P_HAP2	P_NAME	P_EN	P_G	M_HAP1	M_HAP2	M_NAME	M_EN	M_G	
Chr06	15115882		A	C	0	1	block7314		H	7.18019122824675		1	0	block7316
Chr06	15116384		A	G	0	1	block7314		H	7.18019122824675		1	0	block7316
Chr06	15116392		T	A	0	1	block7314		H	7.18019122824675		1	0	block7316
Chr06	15116410		T	G	0	1	block7314		H	7.18019122824675		1	0	block7316
Chr06	15116452		C	T	1	0	block7314		H	7.18019122824675		0	1	block7316
Chr06	15116453		T	C	0	1	block7314		H	7.18019122824675		1	0	block7316
Chr06	15116482		A	T	0	1	block7314		H	7.18019122824675		1	0	block7316
Chr06	15116495		A	G	0	1	block7314		H	7.18019122824675		1	0	block7316
Chr06	15116503		C	T	0	1	block7314		H	7.18019122824675		1	0	block7316
Chr06	15116505		G	T	0	1	block7314		H	7.18019122824675		1	0	block7316
Chr06	15116565		A	T	0	1	block7314		H	7.18019122824675		1	0	block7316
Chr06	15116566		A	G	0	1	block7314		H	7.18019122824675		1	0	block7316
Chr06	15116581		T	A	0	1	block7314		H	7.18019122824675		1	0	block7316
Chr06	15116606		C	T	0	1	block7314		H	7.18019122824675		1	0	block7316
Chr06	15116610		T	G	0	1	block7314		H	7.18019122824675		1	0	block7316
Chr06	15116611		G	T	0	1	block7314		H	7.18019122824675		1	0	block7316
Chr06	15116623		A	C	0	1	block7314		H	7.18019122824675		1	0	block7316
Chr06	15116624		A	G	0	1	block7314		H	7.18019122824675		1	0	block7316
Chr06	15116637		T	C	0	1	block7314		H	7.18019122824675		1	0	block7316

From left to right by column:

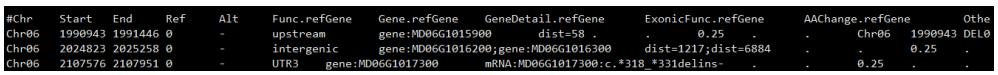
Chr : The chromosome of the SNVs
Start: The start position of the SNVs in the chromosome
End: The end position of the SNVs in the chromosome
Ref: The reference allele
Alt: The alternative allele
Func.refGene: The functional region of the SNVs
Gene.refGene: The closet gene with the SNVs
GeneDetail.refGene: The detail information of the gene
ExonicFunc.refGene: The exonic functional annotation of the SNVs
AAChange.refGene: The amino acid alteration

Otherinfo： Other information

CHROM: The chromosome of the marker
POS: The position of the SNV in chromosome
REF: The reference allele
ALT: The alter allele
P_HAP1: allele in haplotype1 [0 for reference allele; 1 for alter allele] of pollen parent
P_HAP2: allele in haplotype2 [0 for reference allele; 1 for alter allele] of pollen parent
P_NAME: haplotype name in pollen parent
P_EN: the P_HAP2 enriched in H/L pool
P_G: averaged G' value in the P haplotype
M_HAP1: allele in haplotype1 [0 for reference allele; 1 for alter allele] of maternal parent
M_HAP2: allele in haplotype1 [0 for reference allele; 1 for alter allele] of maternal parent
M_NAME: haplotype name in maternal parent
M_EN: the M_HAP2 enriched in H/L pool
M_G: averaged G' value in the M haplotype

*.sv [FILE]

screened SNVs based on genetic rules located in the QTL regions (*:QTL accession)



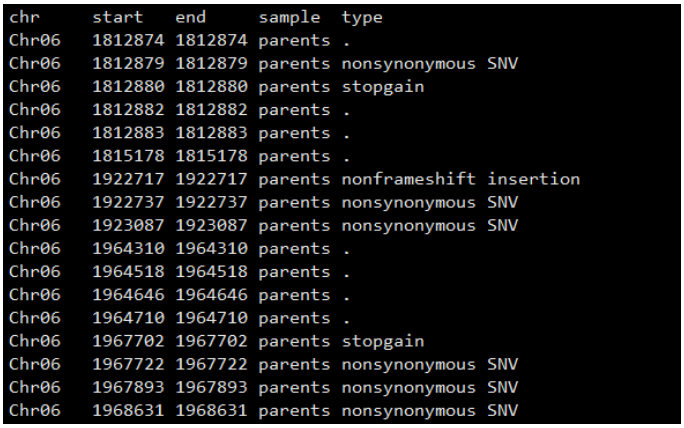
#Chr	Start	End	Ref	Alt	Func.refGene	Gene.refGene	GeneDetail.refGene	ExonicFunc.refGene	AAChange.refGene	Othe
Chr06	1990943	1991446	0	-	upstream	gene:MD06G1015900	dist=58	.	.	0.25	.	.	Chr06	1990943	DEL0
Chr06	2024823	2025258	0	-	intergenic	gene:MD06G1016200;gene:MD06G1016300	dist=1217;dist=6884	.	.	0.25	.	.			
Chr06	2107576	2107951	0	-	UTR3	gene:MD06G1017300	mRNA:MD06G1017300:c.*318_*331delins-	.	.	0.25	.	.			

From left to right by column:

Chr : The chromosome of the SVs
Start: The start position of the SVs in the chromosome
End: The end position of the SVs in the chromosome
Ref: The reference allele
Alt: The alternative allele
Func.refGene: The functional region of the SVs
Gene.refGene: The closet gene with the SVs
GeneDetail.refGene: The detail information of the gene
ExonicFunc.refGene: The exonic functional annotation of the SVs
AAChange.refGene: The amino acid alteration
`Otherinfo:` `Other information`

*.snv.igv.mut [FILE]
screened SNVs based on genetic rules located in the QTL regions (*:QTL accession) [MUT format]

https://software.broadinstitute.org/software/igv/MUT



chr	start	end	sample	type
Chr06	1812874	1812874	parents	.
Chr06	1812879	1812879	parents	nonsynonymous SNV
Chr06	1812880	1812880	parents	stopgain
Chr06	1812882	1812882	parents	.
Chr06	1812883	1812883	parents	.
Chr06	1815178	1815178	parents	.
Chr06	1922717	1922717	parents	nonframeshift insertion
Chr06	1922737	1922737	parents	nonsynonymous SNV
Chr06	1923087	1923087	parents	nonsynonymous SNV
Chr06	1964310	1964310	parents	.
Chr06	1964518	1964518	parents	.
Chr06	1964646	1964646	parents	.
Chr06	1964710	1964710	parents	.
Chr06	1967702	1967702	parents	stopgain
Chr06	1967722	1967722	parents	nonsynonymous SNV
Chr06	1967893	1967893	parents	nonsynonymous SNV
Chr06	1968631	1968631	parents	nonsynonymous SNV

From left to right by column:

Chr: chromosome
start: start location (location of the first base pair in the mutated region)
end: end location (location of the last base pair in the mutated region)
sample: sample or patient ID
type: mutation type (for example, Synonymous, Missense, Nonsense, Indel, etc.)

*.snv.igv.vcf [FILE]

screened SNVs based on genetic rules located in the QTL regions (*:QTL accession) [VCF format]

https://samtools.github.io/hts-specs/VCFv4.2.pdf

*.sv.igv.mut [FILE]

The format is the same as *.snv.igv.mut

*.sv.igv.vcf [FILE]

screened SVs based on genetic rules located in the QTL regions (*:QTL accession) [VCF format]

The format is the same as *.snv.igv.vcf

The command **'igv'** is used to generate files for Integrative Genomics Viewer

```
igv_dir[DIR]
      |reference.fasta [FILE]
      |gene.gtf [FILE]
      |*.sv.igv.mut [FILE]
      |*.snv.igv.mut [FILE]
      |*.snv.igv.vcf  [FILE]
      |*.sv.igv.vcf   [FILE]
      |P.igv          [FILE]
      |PM.igv         [FILE]
      |M.igv          [FILE]
      |snv.vcf        [FILE]
      |snv.maf        [FILE]
      |sv.vcf         [FILE]
      |sv.maf         [FILE]
```

reference.fasta [FILE]

Reference genome

gene.gtf [FILE]

Gene annotation file

*.snv.igv.mut [FILE]
screened SNVs based on genetic rules located in the QTL regions (*:QTL accession) [MUT format]

https://software.broadinstitute.org/software/igv/MUT

*.sv.igv.mut [FILE]
screened SVs based on genetic rules located in the QTL regions (*:QTL accession) [MUT format]

https://software.broadinstitute.org/software/igv/MUT

*.snv.igv.vcf [FILE]

screened SNVs based on genetic rules located in the QTL regions (*:QTL accession) [VCF format]

*.sv.igv.vcf [FILE]

screened SVs based on genetic rules located in the QTL regions (*:QTL accession) [VCF format]

snv.vcf [FILE]

The filtered SNVs between two parents [VCF format]

snv.maf [FILE]

The filtered SNVs between two parents [MAF format]

https://software.broadinstitute.org/software/igv/MutationAnnotationFormat

sv.vcf [FILE]

The filtered SVs between two parents [VCF format]

sv.maf [FILE]

The filtered SVs between two parents [MAF format]