Appendix A

Description of the Program Implemented

For implementation of the GA (Genetic Algorithm), Exhaustive and Conditional search procedures to detect epistatic QTLs was adjusted the interval regression model with interaction effect between two loci (expression 1), assuming uncorrelated, normal and homocedastic errors.

Firstly, by reading phenotypic and genotypic data from n F2 individuals and the genome size (compgenoma variable) is possible to calculate the values of SSE, AIC and BIC for the Exhaustive, Conditional and GA procedure. R resources was used for the implementation. The program file was called "programagratos.txt".

A.1 – Step 1

From simulated or real data set is expected four files as follow:

• Traço: file with quantitative trait data (phenotypic variable) for n F2 individuals. For instance: "tracosbpsratos.txt;

• Marcadores: array that stores the data of all markers coded 0, 1 and 2 (complete data) and -1, 10 and 12 (incomplete), for n F2 individuals. "For instance: "marcadoresratos.txt";

• Existemarcador: vector that stores the values of a binary variable (0 for positions with observed marker information (known genotype data) and 1 otherwise, i.e., if the position is imputed within an interval flanked by observed markers (unknown genotype but to be predicted). For instance: "existemarcadorratos.txt";

• Chrmarc: matrix with the first column defining the length of each chromosome on the map and in the second column the number of markers in each of these chromosomes. For instance: "chrmarcratos.txt".

A.2 – Step 2

From the four files described in step 1 is built "chrpos", the file with the matrix that holds genome positions in the first column (from 1 to 1 cM covering the total length of the genome - compgenoma variable); in column 2 the chromosome number and in column 3 the position (in cM) of the locus within the chromosome.

A.3 – Step 3

Combine matrix chrpos with existemarcador, the vector that is in the fourth column. This matrix is called vetoralg.

A.4 – Step 4

In this step is performed the routine that randomizes uniformly a pair of positions in the genome (ranging from 1 to compgenoma) and such two selected locations are stored in mcp file. If the two selected positions belong to the same chromosome then is tested if the distance between them is at least 10cm (≤ 10). If this distance is not satisfied, other two positions are randomized.

A.5 – Step 5

Table A.1 – Conditional probabilities associated to genotypes (pcag) for a QTL in the interval between markers M_1 e M_2 in a F_2 population.

Genótipo	QQ	Qq	qq
$M_1M_1M_2M_2$	$\frac{(1-r_1)^2(1-r_2)^2}{(1-r)^2}$	$\frac{2r_1(1-r_1)r_2(1-r_2)}{(1-r)^2}$	$\frac{(r_1)^2(r_2)^2}{(1-r)^2}$
$M_1M_1M_2m_2$	$\frac{2(1-r_1)^2 r_2(1-r_2)}{2r(1-r)}$	$\frac{2r_1(1-r_1)\left((r_2)^2+(1-r_2)^2\right)}{2r(1-r)}$	$\frac{2(r_1)^2 r_2(1-r_2)}{2r(1-r)}$
$M_1M_1m_2m_2$	$\frac{(1-r_1)^2(r_2)^2}{r^2}$	$\frac{2r_1(1-r_1)r_2(1-r_2)}{r^2}$	$\frac{(r_1)^2(1-r_1)^2}{r^2}$
$M_1m_1M_2M_2$	$\frac{2r_1(1-r_1)(1-r_2)^2}{2r(1-r)}$	$\frac{2\left((r_1)^2+(1-r_1)^2\right)r_2(1-r_2)}{2r(1-r)}$	$\frac{2r_1(1-r_1)(r_2)^2}{2r(1-r)}$
$M_1m_1M_2m_2$	$\frac{4r_1(1-r_1)r_2(1-r_2)}{2\left(r^2+(1-r)^2\right)}$	$\frac{2\left((r_1)^2+(1-r_1)^2\right)\left((r_2)^2+(1-r_2)^2\right)}{2\left(r^2+(1-r)^2\right)}$	$\frac{4r_1(1-r_1)r_2(1-r_2)}{2\left(r^2+(1-r)^2\right)}$
$M_1m_1m_2m_2$	$\frac{2r_1(1-r_1)(r_2)^2}{2r(1-r)}$	$\frac{2\left((r_1)^2+(1-r_1)^2\right)r_2(1-r_2)}{2r(1-r)}$	$\frac{2r_1(1-r_1)(1-r_2)^2}{2r(1-r)}$
$m_1m_1M_2M_2$	$\frac{(r_1)^2(1-r_2)^2}{r^2}$	$\frac{2r_1(1-r_1)r_2(1-r_2)}{r^2}$	$\frac{(1-r_1)^2(r_2)^2}{r^2}$
$m_1m_1M_2m_2$	$\frac{2(r_1)^2 r_2(1-r_2)}{2r(1-r)}$	$\frac{2r_1(1-r_1)\left((r_2)^2+(1-r_2)^2\right)}{2r(1-r)}$	$\frac{2(1-r_1)^2 r_2(1-r_2)}{2r(1-r)}$
$m_1m_1m_2m_2$	$\frac{(r_1)^2(r_2)^2}{(1-r)^2}$	$\frac{2r_1(1-r_1)r_2(1-r_2)}{(1-r)^2}$	$\frac{(1-r_1)^2(1-r_2)^2}{(1-r)^2}$

Table A.2. Values for the additive and dominance effects for all possible genotype.

Genótipo	Xa	Xd
$M_1M_1M_2M_2$	pcag[1,1]-pcag[1,3]	pcag[1,2]
$M_1M_1M_2m_2$	pcag[2,1]-pcag[2,3]	pcag[2,2]
$M_1M_1m_2m_2$	pcag[3,1]-pcag[3,3]	pcag[3,2]
$M_1m_1M_2M_2$	pcag[4,1]-pcag[4,3]	pcag[4,2]
$M_1m_1M_2m_2$	pcag[5,1]-pcag[5,3]	pcag[5,2]
$M_1m_1m_2m_2$	pcag[6,1]-pcag[6,3]	pcag[6,2]
$m_1m_1M_2M_2$	pcag[7,1]-pcag[7,3]	pcag[7,2]
$m_1m_1M_2m_2$	pcag[8,1]-pcag[8,3]	pcag[8,2]
$m_1m_1m_2m_2$	pcag[9,1]-pcag[9,3]	pcag[9,2]

Values for the additive and dominance effects for all possible genotype conditional probabilities are calculated. For incomplete data (values coded as 10, 12 and -1) it is attributed the values of weights XaXd considering the different possibilities for the missing data of the flanking markers of an individual.

Table A.3 shows the compositions of the different genotypes of markers that flank a particular QTL. The table presents the corresponding weights (in percentage) that are used in the calculation of additive effects (Xa) and dominance effects (Xd), when appropriate, considering incomplete data marker.

It is important to note that for the calculation of Xa on incomplete data, multiply the value of each Xa assuming complete data for its respective weight (in percentage), and finally, takes place the sum of these products for the different possibilities. Repeat the same procedure for calculating the dominance effects, if any.

This estimation method Xa (or Xd if applicable) on incomplete data was based on the proposals of Beerli and Felsenstein (1999) and Felsenstein et al. (1999).

Table A.3. Composition of the genotype of the markers with their respective percentages (weights) for each combination of marker data.

M1	M2	Composition
0	0	$m_1m_1m_2m_2$(100%)
0	1	$m_1m_1M_2m_2$(100%)
0	2	$m_1m_1M_2M_2$(100%)
0	-1	$m_1m_1M_2M_2$(25%) + $m_1m_1M_2m_2$(50%) + $m_1m_1m_2m_2$(25%)
0	10	$m_1m_1M_2m_2$(66,7%) + $m_1m_1m_2m_2$(33,3%)
0	12	$m_1m_1M_2M_2$(33,3%) + $m_1m_1M_2m_2$(66,7%)
1	0	$M_1m_1m_2m_2$ (100%)
1	1	$M_1m_1M_2m_2$(100%)
1	2	$M_1m_1M_2M_2$(100%)
1	-1	$M_1m_1M_2M_2$(25%) + $M_1m_1M_2m_2$(50%) + $M_1m_1m_2m_2$(25%)
1	10	$M_1m_1M_2M_2$(33,3%) + $M_1m_1M_2m_2$(66,7%)
1	12	$M_1m_1M_2m_2$(66,7%) + $M_1m_1m_2m_2$(33,3%)
2	0	$M_1M_1m_2m_2$(100%)
2	1	$M_1M_1M_2m_2$(100%)
2	2	$M_1M_1M_2M_2$(100%)
2	-1	$M_1M_1M_2M_2$(25%) + $M_1M_1M_2m_2$(50%) + $M_1M_1m_2m_2$(25%)
2	10	$M_1M_1M_2m_2$(66,7%) + $M_1M_1m_2m_2$(33,3%)
2	12	$M_1M_1M_2M_2$(33,3%) + $M_1M_1M_2m_2$(66,7%)
-1	0	$M_1M_1m_2m_2$(25%) + $M_1m_1m_2m_2$(50%) + $m_1m_1m_2m_2$(25%)
-1	1	$M_1M_1M_2m_2$(25%) + $M_1m_1M_2m_2$(50%) + $m_1m_1M_2m_2$(25%)
-1	2	$M_1M_1M_2M_2$(25%) + $M_1m_1M_2M_2$(50%) + $M_1m_1M_2M_2$(25%)
-1	-1	$M_1M_1M_2M_2$(6,25%) + $M_1M_1M_2m_2$(12,5%) + $M_1M_1m_2m_2$(6,25%) + $M_1m_1M_2M_2$(12,5%) + $M_1m_1M_2m_2$(25%) + $M_1m_1m_2m_2$(12,5%) + $m_1m_1M_2M_2$(6,25%) + $m_1m_1M_2m_2$(12,5%) + $m_1m_1m_2m_2$(6,25%)
-1	10	$M_1M_1M_2m_2$(16,7%) + $M_1M_1m_2m_2$(33,3%) + $M_1m_1M_2m_2$(16,7%) + $M_1m_1m_2m_2$(16,7%) + $m_1m_1M_2m_2$(16,7%) + $m_1m_1m_2m_2$(8,3%)
-1	12	$M_1M_1M_2M_2$(8,3%) + $M_1M_1M_2m_2$(16,7%) + $M_1m_1M_2M_2$(16,7%) + $M_1m_1M_2m_2$(33,3%) + $m_1m_1M_2M_2$(8,3%) + $m_1m_1M_2m_2$(16,6%)
10	0	$M_1m_1m_2m_2$(66,7%) + $m_1m_1m_2m_2$(33,3%)
10	1	$M_1m_1M_2m_2$(66,7%) + $m_1m_1M_2m_2$(33,3%)
10	2	$M_1m_1M_2M_2$(66,7%) + $m_1m_1M_2M_2$(33,3%)
10	-1	$M_1m_1M_2M_2$(16,7%) + $M_1m_1M_2m_2$(33,3%) + $M_1m_1m_2m_2$(16,7%) + $m_1m_1M_2M_2$(8,3%) + $m_1m_1M_2m_2$(16,7%) + $m_1m_1m_2m_2$(8,3%)
10	10	$M_1m_1M_2m_2$(44,4%) + $M_1m_1m_2m_2$(22,2%) + $m_1m_1M_2m_2$(22,2%) + $m_1m_1m_2m_2$(11,1%)
10	12	$M_1m_1M_2M_2$(22,2%) + $M_1m_1M_2m_2$(44,4%) + $M_1m_1m_2m_2$(11,1%) + $m_1m_1M_2m_2$(22,2%)
12	0	$M_1M_1m_2m_2$(33,3%) + $M_1m_1m_2m_2$(66,7%)
12	1	$M_1M_1M_2m_2$(33,3%) + $M_1m_1M_2m_2$(66,7%)
12	2	$M_1M_1M_2m_2$(33,3%) + $M_1m_1M_2m_2$(66,7%)
12	-1	$M_1M_1M_2M_2$(8,3%) + $M_1M_1M_2m_2$(16,7%) + $M_1M_1m_2m_2$(8,3%) + $M_1m_1M_2M_2$(16,7%) + $M_1m_1M_2m_2$(33,3%) + $M_1m_1m_2m_2$(16,7%)
12	10	$M_1M_1M_2m_2$(22,2%) + $M_1M_1m_2m_2$(11,1%) + $M_1m_1M_2m_2$(44,4%) + $M_1m_1m_2m_2$(22,2%)
12	12	$M_1M_1M_2M_2$(11,1%) + $M_1M_1M_2m_2$(22,2%) + $M_1m_1M_2M_2$(22,2%) + $M_1m_1M_2m_2$(44,4%)

Next, based on design matrix X it is performed the calculations presented in Table A.4 for each pair of reviewed position.

Table A.4 – Estimators calculated for the regression model.

Resultado	Fórmula
Sum of Squares Total (SST)	$SQT = Y'Y - \frac{\left(\sum_{i=1}^{n} Y_i\right)^2}{n}$
Sum of Squares of Regressão (SQR)	$SSR = (X'X)^{-1} XY - \frac{\left(\sum_{i=1}^{n} Y_i\right)^2}{n}$
Sum of Squares dof Resíduals (SSRe)	$SSE = Y'\left[I - X(X'X)^{-1} X\right]Y$
Coeficient of Determination (R^2)	$CD = \frac{SQR}{SQT}$
Statistics Likelihoods Ratio(LR)	$LR = n \ln\left(\frac{SQT}{SSE}\right)$
Statístics Lod Score (Lod)	$Lod = -n \log_{10}(1 - CD)/2$
Statístics *F*	$F = LR/\Delta n_{par}$
Informação de Akaike Criterion (AIC)	$AIC = -2\ln(LR) + 2\Delta n_{par}$
Informação Bayesiana Criterion (BIC)	$BIC = -2\ln(LR) - \Delta n_{par} \ln(n)$

In Table A.4, *X* is the matrix of additive effects associated with the main and interaction effects for QTL1 and QTL 2, Y is the response vector of the traits (phenotypic responses), *n* is the sample size; Δn_{par} is the number of parameters of the regression model (equal 1, in the case of epistatic model). For model adjustment, the full model including the interaction effect ($Y = m + a_1Xa_1 + a_2Xa_2 + a_{12}Xa_1{*}Xa_2 + e$) is compared to reduced model ($Y = m + a_1Xa_1 + a_2Xa_2 + e$), by testing H_0: $a_{12} = 0$, under $a_1 \neq 0$ and $a_2 \neq 0$, with 1 degree of freedom.

For implementation of the conditional search procedure is necessary to perform the test of the additive effect for each candidate position on the map and, therefore, is considered the matrix Xa instead of X. At this stage we compare the full model ($Y = m + a_1X_{a1} + e$) with the reduced model ($Y = m + e$), by testing only the additive effect (H_0: $a_1 = 0$), with 1 degree of freedom, explained in the previous paragraph.

The output of this stage of the program returns the values of the objective functions SSE, AIC and BIC, adopted to assess the fit of the different models of interaction defined for pairs of positions of QTL 1 and QTL 2 located on the marker map. The values of the SSE, AIC and BIC measures are used in the next step, i.e., in the genetic algorithm.

A.6 – Step 6

This step deals with the implementation of routines for genetic algorithm that is initialized by reading the following parameters: number of solutions (ns), number of generations determined by solution (ng) and mutation probability (pm). The routine follows the steps:

Step 1 - Randomization.

Uniformly four pairs of positions are randomized which are evaluated by the objective functions resulting in an adjustment value for each pair of selected positions (AIC, BIC or SSE) and, additionally, another pair of positions is randomized that initializes the routine with high and absurd value so that the adjustment can be discarded and later booted a generation.

Step 2 - Matrix Boot.

This matrix consists of four lines, each line representing a pair of positions with their corresponding location in the genome, on the chromosome and the setting value.

Step 3 - Selection for tournaments.

Here are two selections performed by the tournament. The first match is made for the values of the first and the second rows of the matrix boot, and the second match is made between the settings of the third and fourth row. The winners (best fitted between the two candidates, i.e., showing the lower SSE , AIC or BIC according the objective function being used in the study) are made by the winner of each confrontation, which follow for the second tournament selection. The winner of this second confrontation adjustment will be updated in the next step (recombination or mutation).

Step 4 - Recombination or Mutation.

This enables the algorithm to decide between recombination and mutation to update the winner setting. It is generated a random variable under an uniform U [0,1] distribution and if

the value of this variable is less than p (fixed value for recombination), the algorithm BLX-α performs recombination, otherwise executes the procedure of mutation.

If the decision to perform the procedure recombination occurred, the algorithm will perform the procedure BLX-α recombination as follow:

a) Generate a random variable under the distribution U[0,1], called h;

b) Generate two random variables under the distribution U[h -1, h + 1], called vector beta;

c) Compute the updated pair of positions that won the second tournament as follow:

pos1'(first updated position) = pos1 + beta[1,1] (pos2 - pos1) and

pos2'(updated second position) = pos2 + beta [2,1] (pos2 - pos1) .

If one or the two current positions result in values less than zero, greater than the length of the genome or the module of the difference between these positions is smaller than 10, the recombination process is repeated from step a) until they get a couple of feasible positions upgraded to obey the restrictions, i.e., a couple of positions that is doable.

If the algorithm follows the procedure of mutation, the next step of the algorithm is to select between drastic or mild mutation, and it is simulated a random variable, called pmlmd, under the U[0,1] distribution , and if pmlmd is less than or equal to 0.5, the procedure proceeds to drastically limit or otherwise the procedure will follow for changing mild.

If the algorithm follows a drastic change, it is performed the procedure for changing the limit, in which point the algorithm will choose between replacing the first element of the first pair of positions (lower limit) and the second element of the pair by compgenoma (upper limit) , both with the same probability of occurrence .

If the algorithm choose to perform the procedure for changing mild it is generated a random variable DELTA under the U[0.5,compgenoma + 0.5] distribution, and then, the algorithm will check between the module and POS1 difference DELTA and pos2 and DELTA and DELTA value overrides the value of the pair of positions to be updated that is closest to DELTA.

Thus, in general, in this step positions are updated by a process called recombination α BLX, with pr probability, or by mutation, with probability equal pm = 1-pr. If in a given

generation it is selected the option of changing, there is a choice of mutation limit (sharp) or mild mutation (uniform), also by generating an U [0,1], with equivalent choice probabilities for both types of mutation , namely 0.5. If the program has opted for drastic change, the program will decide between lower limit and upper limit also from the results of a new U[0,1] generated .

Step 5 - Update

The value that is obtained by objective function for the pair of positions maintained in recombination or mutation process is compared with the four stored results of the boot array, if the "new value" is greater than the "worst" existing value, new update is made. In the case of a "new value" be less than one of the 4 stored solutions it replaces the higher value of the four pairs of positions and a new generation is initialized. Always at the end of a generation the pair of positions with the lowest objective function value is stored as a solution.

The algorithm stops after serving a fixed number of generations and solutions.

A.7 - Step 7 – Exhaustive search

It is calculated the adjustment of the interaction model for all pairs of positions of the genome obeying the constraint that the distance between the two positions considered to be greater than 10cm. The best fit in this case is for the pair of positions that produces the lowest value of the objective function (SSE, AIC and BIC), which will be considered the global optimum point. In despite of the advantage of this procedure to be able to consider all possibilities, it can become computationally infeasible as the genome size increases.

A.8 – Conditional search

It is calculated the setting (SSE, AIC or BIC) to each position individually (single locus model) and then for the position with the best fitted model (say QTL1) it is

calculated the re-adjustment model for evaluating all epistatic model including the remaining positions. The best fit of interaction in this case is the result of the comparison of the models that include QTL1.