QTLEMM: An R Package for QTL mapping and hotspot detection

by Ping-Yuan Chung, You-Tsz Guo, Hsiang-An Ho, Hsin-I Lee, Po-Ya Wu, Man-Hsia Yang, Miao-Hui Zeng, and Chen-Hung Kao

Abstract Statistical methods for QTL mapping and QTL hotspot detection have been well developed and applied to the exploration of the genetic architecture of quantitative traits across various biological studies. In this paper, we introduce **QTLEMM**, an R package designed to implement commonly used statistical methods for QTL mapping and QTL hotspot detection. The **QTLEMM** package offers statistical functions for simulating or analyzing data, computing significance thresholds, and visualizing results of QTL mapping and QTL hotspot detection. For QTL mapping, the **QTLEMM** package offers a variety of functions to perform tasks such as estimating QTL parameters using single-QTL or multiple-QTL methods. These methods encompass linear regression, permutation tests, normal mixture models, and truncated normal mixture models. The package enables the fitting and comparison of numerous statistical models during the QTL mapping process, and also employs Gaussian stochastic processes to compute significance thresholds for QTL detection in genetic linkage maps across diverse experimental populations, including BC, F2, RI, AI, IRI, and IF2 populations. Moreover, the **QTLEMM** package accommodates both complete genotyping and selective genotyping data from various experimental designs for QTL detection. For QTL hotspot detection, the **QTLEMM** package utilizes a permutation algorithm that randomly shifts elements in the QTL matrix with trait grouping to detect QTL hotspots. By trait grouping, it can take trait correlations into account to mitigate the underestimation of hotspot thresholds in the analysis. Moreover, this approach can deal with both individual-level and summarized data, and also identify various types of hotspots at a very low computational cost during the detection process. Our paper offers a comprehensive overview of **QTLEMM**'s primary functions, supported by numerical analyses and graphical outputs. This provides researchers with statistical tools of QTL mapping and hotspot detection to facilitate the discovery of more significant results in the analysis of networks among genes, QTL hotspots and quantitative traits in broad areas of biological studies.

1 Introduction

Many biologically and economically important traits in organisms are quantitative rather than qualitative. These include traditional traits (such as yields and quality in rice, weight and body fat percentage in animals, and diabetes and hypertension in humans) and molecular traits (such as gene expression and protein levels). Quantitative traits typically exhibit continuous variation in a population, so there is no easy way to categorize them. They are likely to be affected by numerous genes each with modest effects and easily affected by environmental factors. Consequently, traditional methods such as the Mendelian segregation ratio analysis, mean and variance analyses, covariance studies, and the examination of familial correlations are very difficult to detect the individual genes contributing to these traits. The genes responsible for quantitative traits are referred to as quantitative trait loci (QTL). For a long time, researchers have tried to obtain individual QTL information for exploring the genetic mechanisms underlying quantitative traits and further to manipulate them for improving the traits. With the availability of fine-scale genetic marker data along the genomes for various organisms, it has become possible to systematically map for and detect individual QTL (QTL mapping) by using more sophisticated statistical methods.

Statistical methods for QTL mapping have been well established (Lander and Botstein 1989; Haley and Knott 1992; Zeng 1993, 1994; Jansen 1993; Xu and Atchley 1995; Kao, Zeng, and Teasdale 1999; Kao 2000, 2004, 2006; Sen and Churchill 2001; Broman et al. 2003; Kao and Zeng 2002; Kao and Zeng 2009, 2010; Kao and Ho 2012; Lee, Ho, and Kao 2014). These methods analyze the marker and trait data from well-designed experimental populations to estimate the QTL parameters, including the numbers, positions, various gene effects (additive, dominance, and interactive), variance components, heritabilities, etc. The experimental populations include the most commonly used populations, such as the backcross and F2 populations, and other more advanced populations, such as recombinant inbred (RI) populations, advanced intercrossed (AI) populations, intermated recombinant inbred (IRI) populations, and immortalized F2 (IF2) populations. The QTL mapping data typically comprise two parts: a set of phenotypic traits of interest and a set of genetic marker genotypes aligned with a fine-scale genetic marker map, obtained from the individuals within an experimental population. The statistical methods are applied to analyze the QTL mapping data and tackle the several central issues, including the estimation of QTL parameters, determination of threshold values and selective genotyping, in the QTL mapping studies. These study has provided important insights into the genetic

mechanisms governing quantitative traits in various organisms, such as rice, maize, alfalfa, Atlantic salmon, trout, etc. (Vaughan, Balazs, and Heslop-Harrison 2007; Chen et al. 2021; Kumar et al. 2024; Meng et al. 2024; Mackay and Anholt 2024).

QTL hotspots, characterized by genomic locations enriched in QTL, represent a common and notable feature when collecting many QTL for various traits across various biological studies (Chardon et al. 2004; West et al. 2007; Breitling et al. 2008; C. Wu et al. 2008; Yang, Wu, and Kao 2019; Meng et al. 2024). These hotspots are significant and appealing due to their high informativeness and potential harboring for genes related to quantitative traits. Presently, both the genetical genomics experiments and public QTL databases can provide the data sets with numerous QTL for hotspot analysis. The genetical genomics experiment provides individual-level data, which includes original marker genotypes and many molecular traits. This allow to detect thousands of QTL in a single experiment. On the other hand, public databases such as GRAMENE, Q-TARO, Rice TOGO browser, PeanutBase, and MaizeGDB curate thousands of summarized QTL data from various independent QTL experiments that contain detected QTL, trait names, and reference sources without any individual-level data. Statistical methods using either type of data for detecting QTL hotspots have been proposed, and they are mainly based on the permutation test approach (C. Wu et al. 2008; Li, Lu, and Cui 2010; Breitling et al. 2008; Neto et al. 2012; Yang, Wu, and Kao 2019; P.-Y. Wu, Yang, and Kao 2021). Among these methods, the statistical framework outlined by Yang, Wu, and Kao (2019) and P.-Y. Wu, Yang, and Kao (2021) has the notable features of being able to handle both types of the data and save computational cost in the detection of QTL hotspots.

We provide a comprehensive overview of the primary R functions in the **QTLEMM** package. These functions can implement some commonly used and popular statistical methods of QTL mapping and QTL hotspot detection to analyze the data from various experimental populations for exploring the genetic architecture of quantitative traits. The package also offers functions that can simulate QTL mapping data for the purpose of simulation study. Results from analyses are presented through numerical and graphical outputs, facilitating interpretation and visualization of findings. The **QTLEMM** package provides researchers with statistical tools to explore the network among expressivity of genes, QTL hotspots, and quantitative traits in genes, genomes, and genetics studies.

2 Methods and Models

Identifying individual QTL (QTL mapping) is a crucial endeavor aimed at understanding the genetic basis and architecture of quantitative traits, thereby facilitating the trait manipulation and improvement. Since the specific locations of QTLs are unknown prior to mapping and they could potentially be located anywhere along the genome, the primary objectives of statistical methods are centered around searching for individual QTLs and subsequently fitting them all into statistical model for the estimation of QTL parameters.

Detection of QTL

Lander and Botstein (1989) were the first to propose a QTL mapping procedure known as interval mapping (IM), which systematically searches the entire genome for QTLs. The IM approach utilizes one marker interval (one flanking marker pair) at a time to establish a putative QTL at a specific position. It models the relationship between a quantitative trait and the putative QTL at that position, subsequently testing for the presence of the QTL by investigating its effects. For a putative QTL, denoted as Q, at a specific fixed position along the genome, the IM model for an individual i with a phenotypic trait value y_i can be expressed as follows:

$$y_i = G_i + \varepsilon_i \tag{1}$$

where G_i represents the genotypic value contributed by the QTL genotype, and ε_i is a residual assumed to follow a normal distribution with mean 0 and variance σ^2. For the individuals in a population derived from two inbred lines, such as the F2 population, the genotypes of their Q can be one of the three possible genotypes, P_1 homozygote (QQ), heterozygote (Qq) and P_2 homozygote (qq). Various genetic models have been proposed to characterize the relationship between genotypic values and gene effects (Cockerham 1954; Van Der Veen 1959; Weir and Cockerham 1977; Kao and Zeng 2002). According to Cockerham's model (Kao and Zeng 2002), the relationship between the three genotypic values and the QTL effects can be modeled as $G_{QQ} = \mu + a - d/2$, $G_{Qq} = \mu + d/2$ and $G_{qq} = \mu - a - d/2$, respectively, where a and d represent the additive and dominance effects of the QTL, respectively. We then can construct an equivalent model based on equation (1) for an individual i as follows:

$$y_i = \mu + ax_i + dz_i + \varepsilon_i \tag{2}$$

where $(x_i, z_i) = (1, -1/2)$, $(0, 1/2)$ or $(-1, -1/2)$ if the QTL genotype of y_i is QQ, Qq or qq. Equation (2) builds the relationship between the genotypic values and QTL genotypes. If the putative QTL is located at the marker, the IM model simplifies to a regression model. However, if the putative QTL is positioned at x within the marker interval (M,N), the genotypes of the QTL are not directly observable and must be inferred from its flanking markers M and N. In this scenario, the statistical model of IM typically becomes a normal mixture model. Given data with n individuals, the likelihood function of the IM model for $\theta = (\mu, a, d, \sigma^2)$ can be expressed as follows:

$$L(\theta|Y, X) = \prod_{i=1}^{n}[\sum_{j=1}^{3} p_{ij} \times f(y_i|\mu_j, \sigma^2)] \tag{3}$$

where $f(y_i|\mu_j, \sigma^2)$ represents a normal probability density function with mean μ_j and variance σ^2. The μ_j's correspond to the genotypic values of the three different QTL genotypes ($\mu_1 = G_{QQ}$,$\mu_2 = G_{Qq}$,$\mu_3 = G_{qq}$), while p_{ij}'s denote the mixing proportions (conditional probabilities) of the three QTL genotypes inferred from the two flanking markers (refer to Kao and Zeng 2009 for obtaining p_{ij}'s in various experimental populations). By treating the normal mixture model as an incomplete-data problem, the EM algorithm (Dempster, Laird, and Rubin 1977) can be readily implemented to obtain the maximum likelihood estimates (MLE) of the parameters. Subsequently, a likelihood ratio test (LRT) can be performed to test the null hypothesis of no QTL ($H_0 : a = 0$ and $d = 0$) at the position x. With a fine-scale genetic marker map throughout the genome, Interval Mapping (IM) can be conducted at all positions covered by markers to produce a continuous LRT statistic profile along chromosomes. By setting a predetermined LRT threshold, the position with the significantly maximum LRT statistic in a chromosome region is considered the estimated QTL location. This method enables the systematic search and identification of QTLs at the genome-wide level, thereby facilitating the estimation of QTL parameters. However, since the search process for QTL needs to be performed at every position of the genome, the iterative expectation-maximization (EM) algorithm can become computationally expensive for QTL mapping (Haley and Knott 1992; Kao 2000). Haley and Knott (1992) introduced regression interval mapping (REG IM) as an approximation to interval mapping (IM), aimed at reducing computational costs. In REG IM, the quantitative trait value is regressed on the conditional expected genotypic value, providing a computationally efficient alternative to full interval mapping (Haley and Knott 1992), although the approximation may not be satisfactory in all cases (Kao 2000; Sen and Churchill 2001).

The IM approach focuses on one putative QTL at a time within the model. However, this method may introduce bias in the identification and estimation of QTLs when multiple QTLs are present in the same linkage group (Lander and Botstein 1989; Haley and Knott 1992; Zeng 1994). To address this issue, composite interval mapping (CIM), which combines interval mapping with multiple regression analysis, was proposed (Jansen 1993; Zeng 1994). In CIM, the method involves using other markers as covariates during the test for a putative QTL. This approach aims to mitigate the interference of other QTLs and reduce residual variance, thereby improving the accuracy of the test. To further enhance QTL mapping, Kao, Zeng, and Teasdale (1999) introduced the multiple interval mapping (MIM) approach. This method aims to leverage multiple marker intervals concurrently to incorporate several putative QTL into the model for QTL mapping. For instance, considering m putative QTL, Q_1, Q_2,..., and Q_m, located at given positions within m separate marker intervals, (M_1,N_1), (M_2,N_2),..., and (M_m,N_m), respectively, the MIM model fitted these m putative QTL can be expressed as follows:

$$y_i = \mu + \sum_{j=1}^{m}(a_j x_{ij} + d_j z_{ij}) + \varepsilon_i \tag{4}$$

For m putative QTL in the model, there are 3^m possible QTL genotypes. The likelihood function of the MIM model for $\theta = (\mu, a_1, d_1, a_2, d_2, ..., a_m, d_m, \sigma^2)$ becomes a 3^m normal mixture model

$$L(\theta|Y, X) = \prod_{i=1}^{n}[\sum_{j=1}^{3^m} p_{ij} \times f(y_i|\mu_j, \sigma^2)] \tag{5}$$

(replacing the number 3 by 3^m in equation (3)), where p_{ij}'s are the conditional probabilities of the 3^m possible QTL genotypes given the flanking marker genotypes. The general formulas by Kao and Zeng (1997), formulated based on the EM algorithm, can be used to estimate the parameters of the m QTL. To avoid using the iterative EM algorithm, alternative approximate methods considering multiple QTL in the model for QTL mapping include REG interval mapping (Haley and Knott 1992) and multiple imputation by Sen and Churchill (2001). While the two approximate methods offer

faster computational speeds, their differences compared to the MIM method in QTL analysis can be significant in certain situations, as discussed by Kao (2000) and Sen and Churchill (2001), and demonstrated through empirical examples. Subsequently, Kao (2004), Kao (2006) and Kao and Zeng (2009) extended the MIM approach to various advanced populations for QTL mapping, considering specific genome structures present in advanced populations. In addition, Lee, Ho, and Kao (2014) further developed the MIM method for the selective genotyping design, a topic we discuss below. The MIM approach indeed offers enhanced precision and power in QTL mapping. It enables the analysis and estimation of epistasis between QTL, more accurate prediction of genotypic values of individuals, and estimation of heritabilities of quantitative traits.

Determination of threshold values

In the interval mapping procedure, a series of null hypotheses, both correlated and uncorrelated, are tested using likelihood ratio test (LRT) statistics across all genomic positions. Given the multiplicity of tests, controlling genome-wide error rates is crucial when determining threshold values for claiming significant QTL detection. It has been recognized that various factors, such as the number and size of intervals, population genome structures, and marker density, are involved and should be considered in determining the threshold value of QTL detection. To address this challenge, several analytical, empirical, and numerical approaches have been proposed to obtain the threshold values. These include methods like Bonferroni adjustment, Ornstein-Uhlenbeck process, numerical simulation, permutation test, and Gaussian process. Each offers unique insights and advantages in obtaining threshold values tailored to the specific characteristics of the QTL mapping study (Lander and Botstein 1989; Churchill and Doerge 1994; Rebai, Goffinet, and Mangin 1994; Piepho 2001; Zou 2004; Chang et al. 2009; Guo 2011; Kao and Ho 2012). In practice, computational efficiency is a crucial consideration when selecting an approach for obtaining threshold values in QTL mapping studies. While numerical methods like permutation tests or numerical simulations may be computationally intensive, analytical methods offer a more efficient alternative with lower computational costs. However, analytical methods often rely on certain assumptions, such as normality, which may not always hold true in practice (Rebai, Goffinet, and Mangin 1994; Piepho 2001; Kao and Ho 2012). The Gaussian process approaches by Chang et al. (2009), Guo (2011) and Kao and Ho (2012) can stand out as particularly efficient, as we found that it is approximately 7700 times faster than the permutation method in obtaining thresholds. This significant improvement in computational speed makes the Gaussian process method a highly attractive option as far as the computational efficiency is concerned in determining the threshold values for QTL detection.

Chang et al. developed a score test for the detection of QTL in the backcross population and showed that the asymptotic distribution of the score test statistics, denoted as $u(x_i)$ for $i = 1, 2, ..., k$, at all the k sequential positions in the genome, follows a Gaussian stochastic process characterized by a mean of zero and a well-structured variance-covariance matrix. Furthermore, as the squared score statistic $u^2(x)$ is asymptotically equivalent to the LRT statistic (Cox and Hinkley 1979; Chang et al. 2009), the distribution of sup $u^2(x)$ along the genome under the null hypothesis can be used to assess the threshold value of the LRT statistic in QTL mapping. Based upon this concept, Guo (2011) and Kao and Ho (2012) extended Chang et al.'s methodology by deriving more general score test statistics and Gaussian processes tailored for evaluating threshold values in various populations, including the F2 population and other advanced populations. These advancements provide researchers with statistical tools to determine the significance thresholds for QTL mapping analyses in diverse experimental populations.

In the scenario of the F2 population, each of the k positions is linked with two score test statistics: one for the additive effect and the other for the dominance effect. Let U represent a vector whose components are the score test statistics at the k genomic positions. Therefore, the vector U has length of $2k$. The asymptotic distribution of U follows a Gaussian stochastic process, denoted as $U\ N(\mu, \Sigma)$, which is a multivariate normal distribution with a probability density function given by:

$$p(U; \mu, \Sigma) = \frac{1}{(2\pi)^{\frac{n}{2}} |\Sigma|^{\frac{1}{2}}} \exp(-\frac{1}{2}(U-\mu)^T \Sigma^{-1}(U-\mu)) \tag{6}$$

Here $\mu = 0$ represents the mean of the distribution, indicating that the score test statistics are centered around zero. The variance-covariance matrix Σ captures the variability and correlations among the score test statistics across different genomic positions. The structure of Σ is intricately linked to the population genome structure and is typically well-defined in experimental populations. The elements of Σ are determined based on the genotypic distributions of one, two, three, and four genes within the population. In backcross and F2 populations, whose genomes have the Markovian structure under the Haldane map function (J. B. Haldane 1919), the genotypic distributions of three and four genes can be derived from the genotypic frequencies of pairwise genes. However, in advanced populations,

the genomes no longer adhere to the Markovian property and are more complex. Consequently, obtaining the genotypic distributions of two, three, and four genes directly becomes challenging in such populations. Indeed, the transition equations proposed by J. Haldane and Waddington (1931), Geiringer (1944), and Kao and Zeng (2010) provide valuable tools for deriving genotypic frequencies of two, three, and four genes, facilitating the construction of the variance-covariance matrix. These equations offer insights into the genotypic distribution of various experimental populations, enabling a deeper understanding of variance-covariance structures between genes. The general frameworks of the score test statistics and Gaussian processes introduced by Guo (2011) and Kao and Ho (2012) can be used to obtain the threshold values of QTL mapping for genomes with different sizes and marker densities in various experimental populations, such as backcross, F2, and more advanced populations. Importantly, these methods have very low computational costs, making them practical for large-scale analyses. In practice, when given a specific significance level and genome size, threshold values should be adjusted to account for denser marker maps and more advanced populations. This adjustment ensures that the statistical analysis appropriately controls for multiple testing and accounts for the complexities inherent in different genetic backgrounds and experimental designs.

Selective genotyping

The cost of conducting QTL mapping experiments includes both phenotyping and genotyping expenses. In situations where budget constraints are not a primary concern, researchers usually choose complete genotyping, wherein all individuals in the sample undergo both genotyping and phenotyping procedures. However, despite recent reductions in genotyping costs, researchers frequently encounter insufficient budgets that prevent them from fully covering the expenses of complete genotyping. In situations where budgets are insufficient, researchers may explore alternative cost-saving approaches. Selective genotyping has been known as a cost-saving strategy to reduce genotyping work and can still maintain nearly equivalent efficiency to complete genotyping in QTL mapping (Lebowitz, Soller, and Beckmann 1987; Lander and Botstein 1989; Xu and Vogl 2000; Lee, Ho, and Kao 2014). This method involves selecting individuals from the high and low extremes of the trait distribution for genotyping, while leaving the remaining individuals ungenotyped within the entire sample. By focusing genotyping on individuals with extreme trait values, researchers can still capture most of the genetic variation in the sample to maintain efficiency. Overall, selective genotyping allows researchers to balance between budget constraints and mapping efficiency in QTL detection analysis.

Suppose that the sample consists of n individuals, out of which n_s individuals with extreme trait values ($n_s/2$ each from the upper and lower extremes) are selected for marker genotyping. The remaining $n_u = n - n_s$ individuals are not genotyped. Statistical QTL mapping methods for analyzing selective genotyping data can either consider all the n individuals (full data) or consider just the n_s genotyping individuals (genotyping data) in their models for QTL detection. If only the genotyping data are utilized in the analysis, data of this sort are called centrally truncated data. Xu and Vogl (2000) and Lee, Ho, and Kao (2014) introduced the truncated model within the mixture framework of interval mapping procedure, presenting a truncated normal mixture model for QTL analysis. For n_s genotyped individuals, the likelihood function for θ in the m QTL model can be expressed as follows:

$$L(\theta|Y,X) = \prod_{i=1}^{n_s}[\sum_{j=1}^{3^m} p_{ij} \times \frac{f(y_i|\mu_j,\sigma^2)}{U_j}] \tag{7}$$

where

$$U_j = \int_{-\infty}^{T_L} f(y_i|\mu_j,\sigma^2)dy_i + \int_{T_R}^{\infty} f(y_i|\mu_j,\sigma^2)dy_i \tag{8}$$

is the cumulative density with trait values greater than T_R (right truncated point) and lower than T_L (left truncated point), such that $P(y_i > T_R) = P(y_i < T_L) = n_s/2n$. Further details on the EM algorithm for obtaining the Maximum Likelihood Estimates (MLE) of the parameters are provided in Lee, Ho, and Kao (2014). If the full data are fitted into the statistical model for QTL analysis, the model likelihood can be expressed as follows:

$$L(\theta|Y,X) = \prod_{i=1}^{n_s}[\sum_{j=1}^{3^m} p_{ij} \times f(y_i|\mu_j,\sigma^2)] \times \prod_{i=1}^{n_u}[\sum_{j=1}^{3^m} q_j \times f(y_i|\mu_j,\sigma^2)] \tag{9}$$

where the first term represents the likelihood for the n_s genotyped individuals, while the second term accounts for the n_u ungenotyped individuals.

Note that p_{ij}'s are derived from the conditional probabilities of the QTL genotypes given their flanking marker genotypes, and q_j's represent the proportions of QTL genotypes in the ungenotyped

individuals (Lee, Ho, and Kao 2014). In the parameter estimation, the same EM algorithm employed for complete genotyping (Kao and Zeng 1997) can be directly applied to obtain the MLE. Studies have indicated that the analysis utilizing full data by model (9) outperforms that utilizing only genotyping data by model (7) because additional information from the ungenotyped individuals is incorporated into the analysis (Xu and Vogl 2000; Lee, Ho, and Kao 2014). Additionally, selective genotyping using larger genotyping proportions, such as $n_s/n = 0.5$, may maintain roughly equivalent power to complete genotyping, whereas using smaller genotyping proportions presents difficulties in achieving the same level of power (Lee, Ho, and Kao 2014). Here, we further extend the models in equations (7) and (9) to map QTL using selective genotyping data from other advanced populations. This extension requires considering the specific genome structures of the advanced populations to compute the proportions p_{ij} for the model in Equation (7) and both the proportions p_{ij} and q_j for the model in Equation (9). The details of the EM algorithm for obtaining the MLE of the parameters in the truncated normal mixture model and the normal mixture model in Equations (7) and (9) are described in Lee, Ho, and Kao (2014).

QTL hotspot detection

Genome-wide QTL hotspot detection typically requires datasets containing numerous QTL to proceed with the analysis. Currently, genetical genomics experiments and public QTL databases serve as two feasible sources of such data. These two data sources have different structures. Genetical genomics experiments provide individual-level data, including original marker genotypes and numerous molecular traits for each individual, enabling the detection of thousands of QTL in a single experiment. On the other hand, public databases such as GRAMENE, Q-TARO, Rice TOGO browser, PeanutBase, and MaizeGDB curate thousands of summarized QTL data. These databases curate the information from numerous independent QTL experiments across various traditional traits, and contain detected QTL, trait names, and reference sources but lack individual-level data. Utilizing both individual-level data from genetical genomics experiments or summarized QTL data from public databases, several statistical methods, primarily based on permutation tests, have been proposed to detect QTL hotspots. West et al. (2007), C. Wu et al. (2008), Li, Lu, and Cui (2010), Breitling et al. (2008) and Neto et al. (2012) have developed statistical methods to detect QTL hotspots for genetical genomics experiments. These methods for detecting QTL hotspots may suffer from several problems, including ignoring the correlation structure among traits, neglecting the magnitude of LOD scores for the QTL, or incurring a very high computational cost. These problems often lead to the detection of excessive spurious hotspots, failure to discover biologically interesting hotspots composed of a small to moderate number of QTL with strong LOD scores, and computational intractability, respectively, during the detection process. Solving these problems is crucial for improving the accuracy and efficiency of QTL hotspot detection.

The statistical framework developed by Yang, Wu, and Kao (2019) and P.-Y. Wu, Yang, and Kao (2021) introduces novel methods to deal with the problems encountered in the approaches of West et al. (2007), C. Wu et al. (2008), Li, Lu, and Cui (2010), Breitling et al. (2008), and Neto et al. (2012). in QTL hotspot detection. Notably, the framework can accommodate both individual-level data from genetical genomics experiments and summarized data from public QTL databases to detect QTL hotspots. By employing trait grouping and top $\gamma_{n,\alpha}$ profile, the framework can also address all the problems at a time for QTL hotspot detection. In trait grouping, the framework utilizes estimated QTL positions instead of phenotypic or genetic correlations among traits to make inference about the tightly linked and/or pleiotropic traits for trait grouping, accounting for the correlation structure among traits. Subsequently, the permutation algorithm introduced by Yang, Wu, and Kao (2019) is applied to randomly shift the tightly linked and/or pleiotropic QTL together along the genome within each trait group. This process can obtain a series of EQF thresholds, denoted as $\gamma_{n,\alpha}$, to facilitate the detection of QTL hotspots during the analysis. The top $\gamma_{n,\alpha}$ threshold is defined as the highest EQF threshold (corresponding to the smallest n) necessary for a bin to qualify as significant for a QTL hotspot within the EQF matrix. In a specific EQF architecture, the top $\gamma_{n,\alpha}$ threshold of a hotspot can be used to assess its significance status relative to others. When assessing a specific hotspot, we can derive several, let's say m, top $\gamma_{n,\alpha}$ thresholds for the m EQF architectures established using m different LOD thresholds. The pattern of the n values within the set of m top $\gamma_{n,\alpha}$ thresholds can outline the dynamic significance status of a hotspot across various EQF architectures. For each hotspot, we profile the top $\gamma_{n,\alpha}$ thresholds and use the profile to outline the LOD-score pattern across the different LOD thresholds. The top $\gamma_{n,\alpha}$ profile can then serve to characterize the types of hotspots with varying sizes and LOD-score distributions, enabling the assessment of small and moderate hotspots with strong LOD scores.

Table 1: List of functions for QTL mapping in the QTLEMM package

Function	Description
EM.MIM()	MIM to estimate the parameters.
EM.MIMv()	MIM to estimate the parameters and their variances.
IM.search()	IM to search for the possible QTL.
MIM.search()	MIM to search for one additional QTL given the identified QTLs in the model.
MIM.points()	MIM to fine tune the QTL parameters by a multidimensional search around the regions of the identified QTL in the model.
EM.MIM2()	MIM to estimate the parameters (for selective genotyping data).
IM.search2()	IM to search for the possible QTL (for selective genotyping data).
MIM.search2()	MIM to search for one additional QTL given the identified QTLs in the model (for selective genotyping data).
MIM.points2()	MIM to fine tune the QTL parameters by a multidimensional search around the regions of the identified QTL in the model (for selective genotyping data).
progeny()	Generate the simulated phenotype and genotype data.
D.make()	Generate the genetic design matrix.
Q.make()	Generate the conditional probability matrix.
LRTthre()	The LRT threshold for QTL detection based on Gaussian stochastic process.

3 QTLEMM for QTL mapping

The **QTLEMM** package comprises functions designed for the statistical QTL mapping analysis. It is capable of handling the data from diverse experimental populations, including BC, F2, RI, and AI populations. For each population, the package considers both complete genotyping data and selective genotyping data for the QTL mapping analysis. The functions within the package enable the utilization of various methods including linear regression, interval mapping (Lander and Botstein 1989), and multiple interval mapping (Kao and Zeng 1997; Kao, Zeng, and Teasdale 1999; Zeng, Kao, and Basten 1999) methods to estimate QTL parameters. The functions for QTL mapping are outlined in Table 1. Below, we demonstrate the application of these functions through QTL mapping analyses on both simulated and real datasets. The `progeny()` function generates simulated phenotype and genotype data for populations based on the specified breeding schemes. These data are then input into the `IM.search()` function to search for potential QTL on the chromosomes. Additionally, the `MIM.search()` function can search for an additional QTL given other identified QTL. The best position can be further obtained by using the `MIM.points()` function. Subsequently, the `D.make()` and `Q.make()` functions are employed to create the genetic design matrix of the QTL effects and the conditional probability matrix of the QTL genotypes, respectively. These two matrices are then utilized in the `EM.MIM()` function to estimate the parameters in the MIM model.

Inputs

In QTL mapping studies, the data typically consist of two components: phenotypic trait values and marker genotypes observed in the individuals under study. To initiate QTL mapping analysis using the **QTLEMM** function, four essential arguments are required: markers ("`marker`"), genotypes ("`geno`"), phenotypes ("`y`") and QTL ("`QTL`"). The "`marker`" argument is a $k * 2$ matrix containing marker information, where k is the number of markers. In the "`marker`" argument, the first column labels the chromosomes where the markers are located, while the second column indicates the marker positions in Morgan (M) or centimorgan (cM). Table 2 provides an example of the "marker" argument, demonstrating that the first three markers of the first chromosome are positioned at 0, 24, and 40 cM, respectively. The "`QTL`" argument is a $q * 2$ matrix containing QTL information, where q is the number of QTLs. Its format is the same as that of the "`marker`" argument. The "`geno`" argument is an $n * k$ matrix containing the marker genotypes of n individuals. Genotypes for P_1 homozygote (MM), heterozygote (Mm) and P_2 homozygote (mm) are encoded as 2, 1 and 0, respectively, while missing genotypes are coded as NA. Table 3 provides an example of the "`geno`" matrix, where each row represents the genotypes of the k markers of an individual. The "`y`" argument is an $n * 1$ vector containing the trait values of n individuals.

Table 2: The format example of marker/QTL information data

chromosome	position_cM
1	0
1	24
1	40
...	...
12	72
12	126

Table 3: The format example of genotype data

	$marker_1$	$marker_2$	$marker_3$	$marker_4$	$marker_5$	...	$marker_k$
ind_1	2	1	1	2	0	...	2
ind_2	2	1	0	0	1	...	1
ind_3	2	2	NA	1	1	...	0
ind_4	0	0	1	0	NA	...	2
...	...	...	...	...	...	...	...
ind_n	1	1	0	0	1	...	0

Operation procedure and simulation example

We offer a simulation example to demonstrate the usage of our R package. Initially, it is necessary to load the **QTLEMM** function and set an arbitrary random number seed, such as 8000, for data simulation in the R environment. The **QTLEMM** function includes all the necessary functions for simulating the data and conducting statistical analysis.

```
library(QTLEMM)
set.seed(8000)
options(digits=3)
```

The `progeny()` function can simulate marker genotype and phenotype data from experimental populations for QTL mapping study. It accepts several key arguments: the `E.vector` argument represents the effects of the QTL; the `ng` argument specifies the generation number; the `h2` argument sets the heritability; the `size` argument contains the sample size; the `type` argument is used to specify the population type, which includes backcross (`type = "BC"`), advanced intercross population (`type = "AI"`), and recombinant inbred population (`type = "RI"`).

Consider the scenario that a simulated dataset consists of 200 F2 individuals with three chromosomes, each with eleven 10 cM equally spaced markers. Three QTLs are positioned at [1,23] (the 23 cM of the 1^{st} chromosome), [1,77] and [2,55], respectively, and their effects are assumed to be -10, 12, and 8, respectively. The 1^{st} and 3^{rd} QTLs have an additive-by-additive effect of 1. The heritability is set at 0.5. The commands used to generate such a dataset are described below. The command of defining the QTL effects is as follows:

```
eff <- c("a1" = -10, "a2" = 12, "a3" = 8, "a1:a3" = 1)
```

If other effects, such as dominance effect of 3 and additive-by-dominance effect of 2, are considered, the arguments in the command is "`d2=3`" and "`a2:d1"=-2`. Please refer to the **QTLEMM** document in CRAN for detailed instructions. The commands for specifying the QTL and marker positions are as follows:

```
marker <- cbind(rep(1:3,each = 11), rep(seq(0, 100, 10), 3))
QTL <- cbind(c(1, 1, 2), c(23, 77, 55))
```

Then, the `progeny()` function can use the above commands to simulate 200 F2 individuals with heritability 0.5.

```
testdata <- progeny(QTL, marker, type = "RI", ng = 2, E.vector = eff, h2 = 0.5, size = 200)
names(testdata)
```

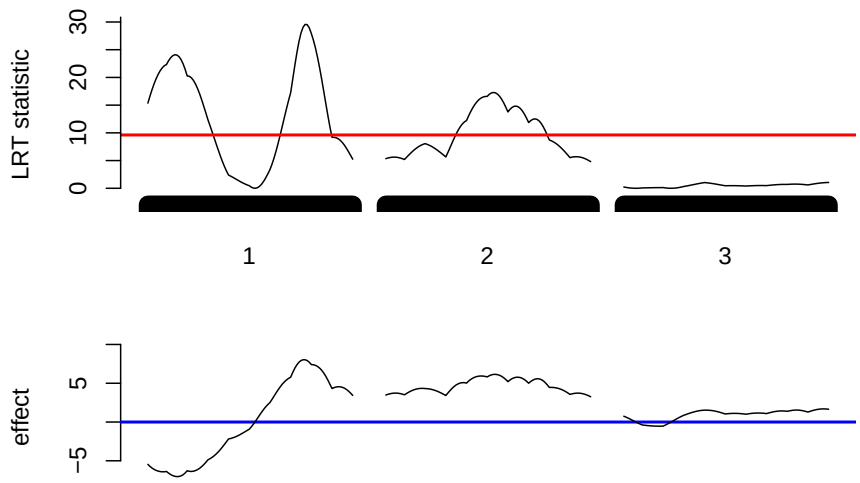

Figure 1: The graphical output generated by the IM.search() function. The upper plot shows the profile of LRT statistics, while the lower plot exhibits the profile of effects. The red line represents the threshold value of 9.62 obtained by using Gaussian process.

```
#> [1] "phe"           "E.vector"      "marker.prog"   "QTL.prog"
#> [5] "genetic.value" "VG"            "VE"

y <- testdata$phe
geno <- testdata$marker.prog
```

The `progeny()` function outputs a dataset into the file named `testdata`. This file contains four parts: phenotypes (`phe`), QTL effects (`E.vector`), marker genotypes (`marker.prog`), and QTL genotypes (`QTL.prog`). The markers and trait values of the 200 individuals in the `testdata` file are further extracted and organized into the `geno` matrix and `y` vector for QTL mapping analysis.

The `IM.search()` function is designed to conduct interval mapping analysis. Its arguments include: the `type` argument specifies the population type (`BC`, `AI`, and `RI` population); the `ng` argument represents the generation number; the speed argument determines the walking speed of the interval mapping analysis (in cM); the `d.eff` argument indicates if the dominant effect will be considered or not (for AI or RI); the `QTLdist` argument specifies the minimum distance (in cM) between the detected QTL; the `plot.all` and `plot.chr` arguments indicate whether plots of the LRT statistic profile will be generated or not. The following is an example of using the `IM.search()` function to perform the interval mapping analysis without considering any dominance effect in the F2 population.

```
IMtest <- IM.search (marker, geno, y, type = "RI", ng = 2, speed = 1, d.eff  =  F,
                     QTLdist  =  15, plot.all = TRUE, plot.chr = FALSE, console = FALSE)


names(IMtest)

#> [1] "effect"        "LRT.threshold" "detect.QTL"

IMtest$LRT.threshold

#>  95%
#> 9.62
```

The outputs of the `IM.search()` function include: estimated effects at all positions (`effect`); LRT threshold (`LRT.threshold`) obtained using Gaussian process; numerical results of the detected QTLs (`detect.QTL`). Figure 1 is the graphical output of the `IM.search()` function. It illustrates the profiles of the LRT statistics and effects across the three chromosomes. There are three significant peaks, indicating three QTLs are detected, on two of the three chromosomes. The LRT threshold obtained using Gaussian stochastic process for assessing the significance of QTL detection is 9.62 in this dataset. The numerical results of the detected QTLs can be listed using the following commands.

```
detQTL <- IMtest$detect.QTL
detQTL
```

```
#>     chr cM    a1  LRT     R2
#> 14    1 14 -7.00 24.1 0.1064
#> 77    1 77  8.03 29.6 0.1324
#> 153   2 53  6.14 17.3 0.0787
```

The IM analysis concludes that the three QTLs are detected at [1,14], [1,77] and [2,53] with effects of -7.00, 8.03 and 6.14, respectively. They contribute approximately 10.64%, 13.24%, and 7.87% of the trait variation, respectively. The analysis of the QTL detection using the IM approach can be further improved using the MIM approach by jointly fitting the three QTL into the MIM model to obtain more precise and accurate estimates of QTL parameters. The `EM.MIM()` function is designed to perform the MIM analysis. Before conducting the `EM.MIM()` function, two matrices, the genetic design matrix (`D.matrix`) and the conditional probability matrix (`cp.matrix`), must be constructed first. The `D.make()` and `Q.make()` functions are utilized to generate the `D.matrix` and `cp.matrix` matrices, respectively. Below are the commands of the `D.make()`, `Q.make()` and `EM.MIM()` functions for the MIM model fitting the three QTL at [1,14], [1,77] and [2,53] with an additive by additive effect (between the QTLs at [1,14] and [2,53]).

The arguments for the `D.make()` function to construct the genetic design matrix of the MIM model fitting the three QTLs are as follows: the first argument is the number of QTL in the MIM model (`nQTL = 3`); the second argument specifies the population type (`type = "RI"`); the arguments a and d indicate whether additive or dominance effects will be considered for the QTL (`a = TRUE`, `d = 0` for considering their additive effects only); the arguments aa, dd, and ad specify the epistatic effects between QTL (`aa = c(1, 3)` for considering the additive by additive effect between the QTLs at [1,14] and [2,53]).

```
dQTL <- detQTL[,1:2]
D.matrix <- D.make(nQTL = 3, type = "RI", a = TRUE, d = 0, aa = c(1, 3))
dim(D.matrix)
```

```
#> [1] 27  4
```

```
head(D.matrix)
```

```
#>     a1 a2 a3 a1:a3
#> 222  1  1  1     1
#> 221  1  1  0     0
#> 220  1  1 -1    -1
#> 212  1  0  1     1
#> 211  1  0  0     0
#> 210  1  0 -1    -1
```

The arguments in the `Q.make()` function for generating the conditional probability matrix of the three-QTL MIM model are as follows.

```
cp.matrix <- Q.make(dQTL, marker, geno, type = "RI", ng = 2)$cp.matrix
dim(cp.matrix)
```

```
#> [1] 200  27
```

Three inputs are required to drive the `EM.MIM()` function for performing the MIM analysis: the genetic design matrix (`D.matrix`); the conditional probability matrix (`cp.matrix`); the phenotypic values (`y`). The outputs from the `EM.MIM()` function include a vector containing the estimated QTL effects (`E.vector`), the mean (`beta`), the residue variance (`variance`), the posterior probabilities matrix (`PI.matrix`), the log likelihood value (`log.likelihood`), the LRT statistics (`LRT`), the coefficient of determination (`R2`), the estimated trait values (`y.hat`), and the iteration time (`iteration.time`).

```
MIMtest <- EM.MIM(D.matrix, cp.matrix, y, console = FALSE)
names(MIMtest)
```

```
#> [1] "E.vector"         "beta"             "variance"         "PI.matrix"
#> [5] "log.likelihood"   "LRT"              "R2"               "y.hat"
#> [9] "iteration.number"
```

```
MIMtest$E.vector
```

```
#>    a1    a2    a3 a1:a3
#> -9.61 10.29  6.35  1.66

MIMtest$log.likelihood

#> [1] -772

MIMtest$R2

#> [1] 0.411
```

The log likelihood of the MIM model fitting the three QTL with epistasis is approximately -772. The estimated QTL effects are approximately -9.61, 10.29 and 6.35 (true values being -10, 12, and 8), respectively, and the estimated epistatic effect is approximately 1.66. The estimated heritability (R2) is 0.411, while the true heritability is 0.50.

The EM.MIMv() function can provide the asymptotic variance-covariance matrix of the QTL parameters. The inputs in the EM.MIMv() function include: QTL information about the QTL effects and positions (QTL); marker information (marker); genotypes (geno); genetic design matrix (D.matrix); conditional probability matrix (cp.matrix), and phenotypic values (y). If the argument cp.matrix is set to NULL, the conditional probability matrix is constructed from the input QTL information and marker information. If the estimated QTL positions coincide with markers, the asymptotic variance-covariance matrix is not available. Below are the arguments of the EM.MIMv() function to produce the variance-covariance matrix for the MIM model fitting the three detected QTL.

```
MIMv <- EM.MIMv(dQTL, marker, geno, D.matrix, cp.matrix = NULL, y, console = FALSE)
names(MIMv)

#>  [1] "E.vector"         "beta"             "variance"         "PI.matrix"
#>  [5] "log.likelihood"   "LRT"              "R2"               "y.hat"
#>  [9] "iteration.number" "avc.matrix"       "EMvar"
```

The avc.matrix is the asymptotic variance-covariance matrix, and the EMvar contains the asymptotic variances of the estimates.

```
round(MIMv$avc.matrix, 3)

#>            QTL1   QTL2   QTL3     a1     a2     a3  a1:a3 variance     X1
#> QTL1      0.015  0.017  0.013 -0.003  0.000  0.014  0.076   -0.073 -0.003
#> QTL2      0.017  0.004 -0.006 -0.023  0.021 -0.003  0.041   -0.191  0.002
#> QTL3      0.013 -0.006  0.065 -0.034  0.035  0.036  0.134   -0.688  0.009
#> a1       -0.003 -0.023 -0.034  1.417 -0.354 -0.091  0.038    1.775  0.006
#> a2        0.000  0.021  0.035 -0.354  1.585 -0.039  0.154   -2.462 -0.096
#> a3        0.014 -0.003  0.036 -0.091 -0.039  1.463  0.257   -1.185 -0.034
#> a1:a3     0.076  0.041  0.134  0.038  0.154  0.257  3.724   -2.787 -0.096
#> variance -0.073 -0.191 -0.688  1.775 -2.462 -1.185 -2.787  179.411  0.030
#> X1       -0.003  0.002  0.009  0.006 -0.096 -0.034 -0.096    0.030  0.650

round(MIMv$EMvar, 3)

#>     QTL1     QTL2     QTL3       a1       a2       a3    a1:a3 variance
#>    0.015    0.004    0.065    1.417    1.585    1.463    3.724  179.411
#>       X1
#>    0.650
```

In EMvar, the asymptotic variances of the estimated mean, QTL positions and effects are 0.015, 0.004, 0.065, 1.417, 1.585, 1.463 and 3.724, respectively. The asymptotic variances of the estimated mean and residual variance are 0.650 and 179.411, respectively.

The MIM.search() function is designed to fitting the detected QTLs into the model to search the genome for other possible QTL. The arguments in the MIM.search() function include the detected QTL (denoted by dQTL2 in this example), marker (for marker information), geno (for genotypes), y (for phenotypes), type (for population type), ng (for the generation number), D.matrix (for the genetic design matrix), speed (for the walking speed in cM), QTLdist (for the minimum distance between

detected QTLs). The outputs of the `MIM.search()` function include information about the estimates of all search positions (`effect`), the best QTL positions with the largest log likelihood (`QTL.best`), and the estimated effects of the best QTL positions (`effect.best`). For demonstration purposes, assume that the two detected QTLs located at [1,14] and [1,77] are fitted into the model to search for the next (third) QTL considering the additive by additive effect (the design matrix will be the same as that in the above `EM.MIM()` function). Below are the commands of the `MIM.search()` function to conduct the search for the third QTL given the two detected QTLs:

```
dQTL2 <- cbind(c(1, 1), c(14, 77))
MIMs <- MIM.search(dQTL2, marker, geno, y, type = "RI", ng = 2, D.matrix = D.matrix,
                   speed = 1, QTLdist = 15, console = FALSE)
names(MIMs)

#> [1] "effect"      "QTL.best"    "effect.best"

MIMs$QTL.best

#>         chromosome position(cM)
#> QTL 1            1           14
#> QTL 2            1           77
#> QTL new          2           54

MIMs$effect.best

#>             a1             a2             a3          a1:a3            LRT
#>         -9.619         10.302          6.385          1.806        145.876
#> log.likelihood             R2
#>       -772.129          0.412
```

The third QTL is detected at the position [2,54] with an estimated effect of approximately 6.385. The log likelihood of the MIM model fitting the three QTLs at [1,14], [1,77] and [2,54] with epistasis is about -772.13. The LRT statistic for testing the significance of the effects jointly is about 145.876. Another function related to the MIM analysis is the `MIM.points()` function, which is developed to fine tune the estimation of QTL parameters by multidimensional search around the detected QTLs. The fine-tuning range around the detected QTL is specified using the scope argument, while the other arguments are the same as those in the `MIM.search()` function. Below is the command of the `MIM.search()` function for performing a three-dimensional search on the 10 cM range of both sides of the three QTL at [1,14], [1,77] and [2,54] (with additive by additive effect).

```
MIMp <- MIM.points(dQTL, marker, geno, y, type = "RI", ng = 2, D.matrix = D.matrix,
                   speed = 2, scope = 10, console = FALSE)
names(MIMp)

#> [1] "effect"      "QTL.best"    "effect.best"

MIMp$QTL.best

#>      chromosome position(cM)
#> [1,]          1           24
#> [2,]          1           75
#> [3,]          2           53

MIMp$effect.best

#>             a1             a2             a3          a1:a3            LRT
#>        -10.846         11.994          6.503          3.725        181.130
#> log.likelihood             R2
#>       -765.371          0.464
```

The results show that the largest likelihood is found to be -765.371 at positions [1,24], [1,75] and [2,53], and the estimated heritability is 0.464. After fine-tuning, the detected positions are closer to the true positions [1,23], [1,77] and [2,55], compared to the estimated positions [1,14], [1,77] and [2,53] before fine-tuning. With these estimates, other composite genetic parameters such as heritability and variance components of a quantitative trait can be estimated. Additionally, the response to selection can be predicted based on these estimates.

The yeast dataset example

The yeast dataset (Brem et al. 2005) consists of 112 backcross individuals with 5740 traits and 1072 markers. We have reprocessed the raw data into a new dataset called `yeast.process`, which can be downloaded from GitHub using the following command:

```
load(url("https://github.com/py-chung/QTLEMM/raw/main/inst/extdata/yeast.process.RDATA"))
```

The `yeast.process` dataset comprises three lists: the list of marker genotypes (`yeast.process$geno`) that contains the marker genotypes of the 112 individuals; the list of trait values (`yeast.process$pheno`) that contains the trait values of the 112 individuals; the list of marker information (`yeast.process$marker`) that includes the marker map (distances) of the 1072 markers of the 16 chromosomes.

```
geno <- yeast.process$geno
marker <- yeast.process$marker
pheno <- yeast.process$pheno
```

P.-Y. Wu, Yang, and Kao (2021) utilized regression interval mapping (Haley and Knott 1992) to conduct QTL mapping analysis of the yeast dataset. For the demonstration of analyzing selective genotyping data, we select the 3590th trait from the dataset and intentionally deleted the marker genotypes of the individuals with medium trait values to produce selective genotyping data in QTL mapping analysis. Specifically, one half of the individuals with extreme trait values (comprising one quarter each from the upper and lower extremes) are chosen to keep their marker genotypes and trait values, and the marker genotypes of the remaining individuals are deleted and ignored in the analysis. Below are the codes for generating the selective genotyping dataset.

```
y0 <- pheno[, 3590]
quantile(y0)

#>     0%    25%    50%    75%   100%
#> -2.372 -0.661  0.000  0.661  2.372

y <- y0[y0>quantile(y0)[4] | y0<quantile(y0)[2]]
yu <- y0[y0 >= quantile(y0)[2] & y0 <= quantile(y0)[4]]
geno.s <- geno[y0 > quantile(y0)[4] | y0 < quantile(y0)[2],]
```

The vector `y` contains the trait values of the individuals with marker genotypes (the upper and lower 25% individuals), and the `geno.s` argument consists of their marker genotypes. The vector `yu` contains the trait values of individuals without marker genotypes. The `IM.search2()` function can perform several selective genotyping QTL mapping methods, which encompass the Lee, Ho, and Kao (2014) model (Lee, Ho, and Kao (2014), `sele.g = "f"`), the truncated model (Lee, Ho, and Kao 2014, `sele.g = "t"`), and the population frequency-based model (Lee, Ho, and Kao 2014, `sele.g = "p"`), to analyze the selective genotyping dataset. If `sele.g = "n"`, the `IM.search2()` function can be used to analyze the complete genotyping data. The followings are the codes of the `IM.search2()` function to analyze the selective genotyping data of the 3590^{th} trait.

```
library(QTLEMM)
set.seed(8000)
IMtest2 <- IM.search2(marker, geno.s, y, yu, sele.g = "f", type = "BC", ng = 1,
                      plot.all = TRUE, plot.chr = FALSE, console = FALSE)
IMtest2$detect.QTL

#>      chr  cM     a1  LRT     R2
#> 626    3  53  0.893 22.0 0.1128
#> 1579   5 112  0.753 16.0 0.0749
#> 4523  13  22 -0.882 21.7 0.1048
```

The random number seed 8000 is used to set up the Gaussian process to compute threshold values for assessing the significance of QTLs. Figure 2 presents the profiles of the LRT statistics and estimated effects along the genome. It shows that three QTL are detected at [3,53], [5,112] and [13,22], respectively. For comparison, we also conduct the complete genotyping analysis for the 3590^{th} trait using the `IM.search()` function. Belows are the codes of the `IM.search()` function for analyzing the complete genotyping data.

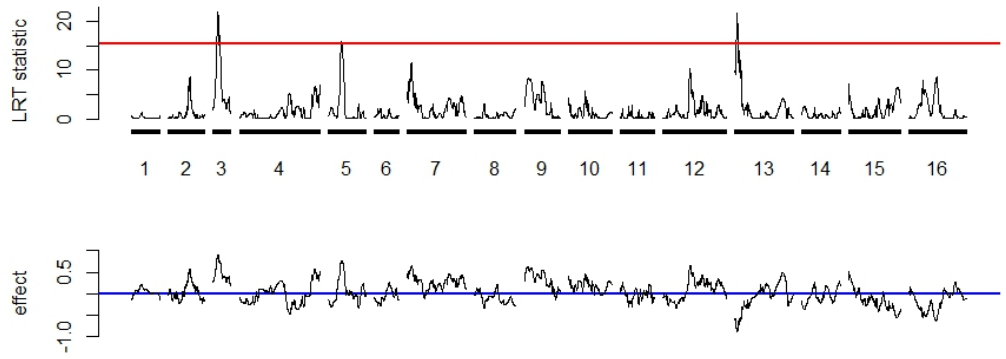

Figure 2: The profiles of the LRT statistics and estimated effects along the genome by using the IM.search2() function to analyze the selective genotyping data of the 3590^{th} trait. The red line indicates the LRT threshold obtained using Gaussian process for evaluating the significance of QTL detection.

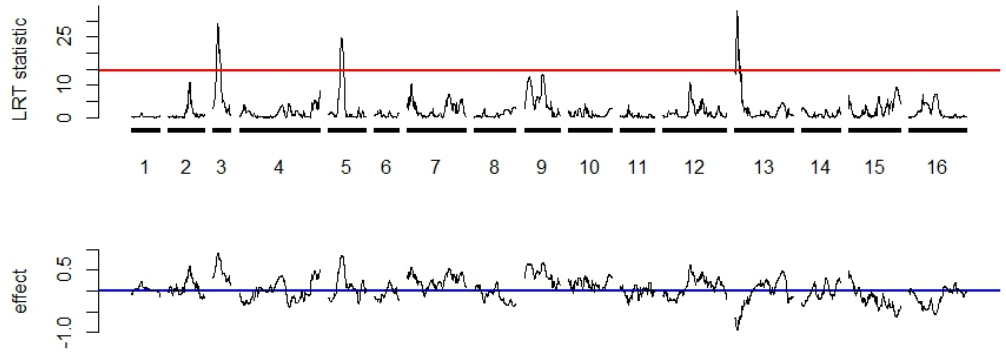

Figure 3: The profiles of the LRT statistics and estimated effects along the genome by using the IM.search() function to analyze the complete genotyping data. The red line indicates the LRT threshold obtained by using Gaussian process for assessing the significance of QTL detection.

```
IMcon <- IM.search(marker, geno, y0, type = "BC", ng = 1, plot.all = TRUE, plot.chr = FALSE,
                   console = FALSE)
IMcon$detect.QTL

#>      chr  cM     a1  LRT    R2
#> 624    3  53  0.904 29.0 0.210
#> 1580   5 117  0.877 25.0 0.171
#> 4511  13  22 -0.945 33.1 0.231
```

The profiles of the LRT statistics and estimated effects along the genomes are presented in Figure 3. It shows that three QTL are detected at [3,53], [5,115] and [13,22], respectively. Both the selective and complete genotyping IM analyses produce similar LRT statistic profiles and estimates of positions and effects. The complete genotyping data have larger LRT statistics and R^2's for each detected QTL.

Using the `IM.search()` and `IM.search2()` functions, the estimates of QTL effects and positions, model likelihoods and model R^2 values were obtained individually. Certainly, we would like to further fit these detected QTLs simultaneously into a multiple-QTL model (the MIM Model). This allows the QTLs to be jointly fitted and controlled in the model to explain more genetic variation of the quantitative traits and obtain more precise estimates. Below are the commands of the `EM.MIM2()` function for performing the selective genotyping MIM model that fits the three detected QTLs and their all possible epistasis.

```
D.matrix <- D.make(3, type = "BC", aa = TRUE)
```

```
dQTL <- IMtest2$detect.QTL[, 1:2]
MIMtest2 <- EM.MIM2(dQTL, marker, geno.s, D.matrix, y = y, yu = yu, sele.g = "p",
                    type = "BC", ng = 1, console = FALSE)
MIMtest2$E.vector

#>     a1     a2     a3  a1:a2  a1:a3  a2:a3
#>  0.751  0.639 -0.762 -0.560  0.279 -0.376

MIMtest2$log.likelihood

#> [1] -124

MIMtest2$LRT

#> [1] 61.4

MIMtest2$R2

#>       [,1]
#> [1,] 0.384
```

The positions of the QTL detected by the selective genotyping MIM model are recorded in the `dQTL` argument. The model R^2 and likelihood are 0.512 and -115.41, respectively. The estimated marginal and epistatic QTL effects are 0.818, 0.744, -0.954, -0.641, 0.371 and -0.423, respectively. The `MIM.points2()` function can be further used to perform a multi-dimensional search around the 5-cM regions of the detected QTL positions ([3,53], [5,112] and [13,22]) to fine-tune the QTL estimates (using the argument of `scope = 5`). Below are the codes of the `MIM.points2()` function for the multi-dimensional search and the fine-tune results.

```
MIMp <- MIM.points2(dQTL, marker, geno.s, y, yu, sele.g = "p", method = "EM",
                    type = "BC", ng = 1, D.matrix = D.matrix, scope = 5, console = FALSE)
MIMp$QTL.best

#>      chromosome position(cM)
#> [1,]          3           58
#> [2,]          5          111
#> [3,]         13           21

MIMp$effect.best

#>             a1             a2             a3          a1:a2          a1:a3
#>          0.665          0.634         -0.780         -0.789          0.591
#>          a2:a3            LRT log.likelihood             R2
#>         -0.389         64.630       -122.497          0.403
```

The model with the largest log likelihood (-113.705) occurs at positions [3,58], [5,111] and [13,21], and the estimated effects are a_1=0.678, a_2=0.758, a_3=-0.964, $i_{a_1a_2}$=-0.866, $i_{a_1a_3}$=0.673, $i_{a_2a_3}$=-0.499, respectively. The model R^2 (estimated heritability) improves from 0.512 to 0.524. Below are the codes of the `MIM.points()` function for analyzing the complete genotyping data of the 3590^{th} trait. After fine-tuning, the estimated positions are at positions [3,54], [5,112] and [13,22], and the estimated effects are a_1=0.703, a_2=0.652, a_3=-0.824, $i_{a_1a_2}$=-0.279, $i_{a_1a_3}$=0.259, $i_{a_2a_3}$=-0.604, respectively. The model R^2 (estimated heritability) is 0.531.

```
dQTLcon <- IMcon$detect.QTL[,1:2]
MIMpcon <- MIM.points(dQTLcon, marker, geno, y0, method = "EM", type = "BC", ng = 1,
                      D.matrix = D.matrix, scope = 5, console = FALSE)
MIMpcon$QTL.best

#>      chromosome position(cM)
#> [1,]          3           54
#> [2,]          5          112
#> [3,]         13           22

MIMpcon$effect.best

#>             a1             a2             a3          a1:a2          a1:a3
#>          0.703          0.652         -0.824         -0.279          0.259
#>          a2:a3            LRT log.likelihood             R2
#>         -0.604        128.497       -114.883          0.531
```

Table 4: List of functions for QTL hotspot detection in the QTLEMM package

Function	Description
LOD.QTLdetect()	Detect QTL by LOD matrix.
EQF.permu()	EQF matrix cluster permutation process for QTL hotspot detection.
EQF.plot()	Depict the EQF plot by the result of permutation process.
Qhot()	This function produces both the numerical and graphical summaries of the QTL hotspot detection in the genomes that are available on the worldwide web including the flanking markers of QTLs.

Table 5: The format of LOD matrix

	bin_1	bin_2	bin_3	bin_4	bin_5	...	bin_n
$trait_1$	0.047	0.116	0.209	0.313	0.342	...	0.358
$trait_2$	0.095	0.176	0.274	0.376	0.301	...	0.342
$trait_3$	0.798	0.67	0.533	0.394	0.342	...	0.284
$trait_4$	0.363	0.321	0.272	0.219	0.192	...	0.149
$trait_5$	0.017	0.01	0.005	0.002	0.001	...	0
...	...	...	...	...	...	...	...
$trait_t$	0.683	0.593	0.471	0.336	0.304	...	0.271

4 QTLEMM for QTL hotspot detection

The analysis of QTL hotspot detection has been a pivotal step towards unraveling the genetic architectures of quantitative traits in the study of genes, genomes and genetics (Breitling et al. 2008; Fu et al. 2009; Neto et al. 2012; Wang et al. 2014; Yang, Wu, and Kao 2019). The genetical genomics experiments and public QTL databases are two feasible sources to provide data with many QTLs for the detection of QTL hotspots. P.-Y. Wu, Yang, and Kao (2021) introduced a statistical framework capable of accommodating both types of data. It addresses various challenges, including handling the correlation structure among traits, identifying different types of hotspots, and ensuring computational efficiency, thereby making it practical for QTL hotspot detection. Below, we present the R-code of the Wu et al. framework by demonstrating the analyses of two real examples: the yeast genetic genomics dataset and the GRAMENE rice database. The functions for QTL hotspot detection of the Wu et al. framework are summarized in Table 4.

The yeast genetic genomics dataset example

There are 5740 molecular traits for the yeast dataset (Brem et al. 2005). The QTL mapping procedure employed for the 3590^{th} trait using the `IM.search()` function can be applied to analyze the remaining 5739 traits, obtaining their LRT statistics across all positions along the genome. These LRT statistics can then be converted into LOD scores using the formula $LOD = LRT/4.6$. Subsequently, the LOD scores are organized into a LOD matrix for QTL hotspot detection, following the methods outlined by Yang, Wu, and Kao (2019) and P.-Y. Wu, Yang, and Kao (2021). The `LOD.QTLdetect()` function is constructed to detect QTL hotspots. It requires two input datasets: the LOD matrix and the bin information on the chromosomes. The LOD matrix is a $t * p$ matrix, where t and p are the numbers of traits and numbers of bins on the chromosomes, respectively. The LOD matrix contains the LOD score at each bin for all traits (refer to Table 5). The bin information is an $n * 2$ matrix, where n is the number of chromosomes, and it contains the information about the bin number on each chromosome. The first column denotes the chromosomes, and the second column denotes the numbers of bins (refer to Table 6).

The LOD matrix of the yeast data can be downloaded from GitHub using the following command. Users can combine the four files (`yeast.LOD.1.RDATA`, `yeast.LOD.2.RDATA`, `yeast.LOD.3.RDATA`, `yeast.LOD.4.RDATA`) to obtain the complete LOD matrix.

```
load(url("https://github.com/py-chung/QTLEMM/raw/main/inst/extdata/yeast.LOD.1.RDATA"))
load(url("https://github.com/py-chung/QTLEMM/raw/main/inst/extdata/yeast.LOD.2.RDATA"))
load(url("https://github.com/py-chung/QTLEMM/raw/main/inst/extdata/yeast.LOD.3.RDATA"))
load(url("https://github.com/py-chung/QTLEMM/raw/main/inst/extdata/yeast.LOD.4.RDATA"))
load(url("https://github.com/py-chung/QTLEMM/raw/main/inst/extdata/yeast.LOD.bin.RDATA"))
LOD <- rbind(yeast.LOD.1, yeast.LOD.2, yeast.LOD.3, yeast.LOD.4)
```

Table 6: The format example of bin information

chromosome	number_of_bin
1	256
2	324
3	160
4	723
...	...
15	463
16	513

```
bin <- yeast.LOD.bin
```

Once the LOD matrix is available, the `LOD.QTLdetect()` function can be applied to detect QTL hotspots. The function's arguments include `LOD` for the LOD matrix (refer to Table 5), `bin` for the numbers of bins on each chromosome (refer to Table 6), `thre` for the threshold value (in terms of LOD) of QTL detection, and `QTLdist` for specifying the minimum distance (cM) between the detected QTL. The numerical results will be output to the `LOD.QTLdetect.result` file.

```
library(QTLEMM)
LOD.QTLdetect.result <- LOD.QTLdetect(LOD, bin, thre = 3, QTLdist = 20,
                                      console = FALSE)

#> step      process

names(LOD.QTLdetect.result)

#> [1] "detect.QTL.number"  "QTL.matrix"         "EQF.matrix"
#> [4] "linkage.QTL.number" "LOD.threshole"      "bin"
```

The `LOD.QTLdetect.result` file is a data list comprising several components: `detect.QTL.number` contains the number of detected QTL of each trait; `QTL.matrix` holds the QTL positions, where elements marked as 1 represent the QTL positions, elements marked as 0 represent bins with LOD scores under the LOD threshold, and other positions are designated as NA; `EQF.matrix` contains the EQF value of each bin; `linkage.QTL.number` indicates the number of linked QTL among all detected QTL; `LOD.threshold` and `bin` remain the same as the input data. With these information, the `EQF.permu()` function embedding the P.-Y. Wu, Yang, and Kao (2021) permutation analysis (with trait grouping) can be applied to detect QTL hotspots. The arguments in the `EQF.permu()` function involve inputting the output data from `LOD.QTLdetect()`, specifying the permutation time (`ptime`), and the type 1 error rate (`alpha`) to carry out the permutation analysis. Additionally, the `Q = TRUE` argument is to perform permutation analysis without trait grouping.

```
result <- EQF.permu(LOD.QTLdetect.result, ptime = 1000, alpha = 0.05, Q = TRUE,
                    console = FALSE)
names(result)

#> [1] "EQF.matrix"           "bin"                  "LOD.threshole"
#> [4] "cluster.number"       "cluster.id"           "cluster.matrix"
#> [7] "permu.matrix.cluster" "permu.matrix.Q"       "EQF.threshold"
```

In the output data, the `EQF.matrix`, `bin`, and `LOD.threshold` lists represent the EQF matrix, bin information matrix, and the LOD threshold respectively, which are the same as those in the input data. The `cluster.number` contains the number of QTLs in each trait group. The `cluster.id` contains the serial number of traits in each trait group. The `cluster.matrix` includes the reduced EQF matrix after trait grouping. The `permu.matrix.cluster` contains the result of permutation with trait grouping, sorted by order. Similarly, the `permu.matrix.Q` contains the result of the permutation without trait grouping (the Q method), also sorted by order. The `EQF.threshold` represents the EQF threshold calculated from the permutation analysis. Moreover, below is the command of the `EQF.plot()` function to provide the EQF architecture of the genome (see Figure 4).

```
EQF.plot(result, plot.all = TRUE, plot.chr = TRUE)
```

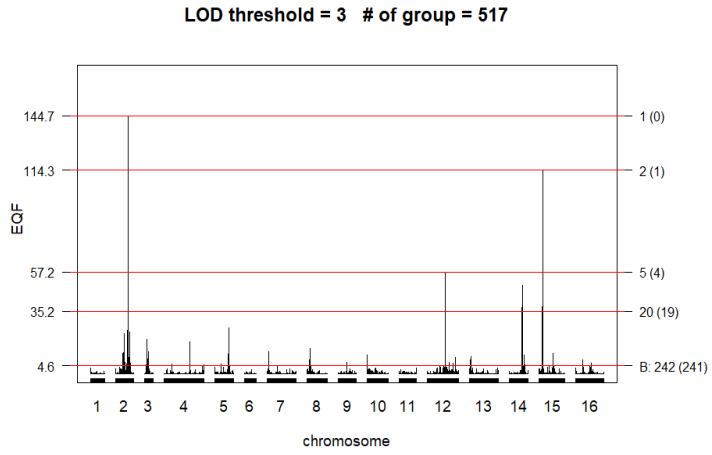

Figure 4: The EQF architecture of all chromosomes by using the EQF.plot() function. The EQF architecture are constructed by the uniform method with bin size of 0.5 cM.

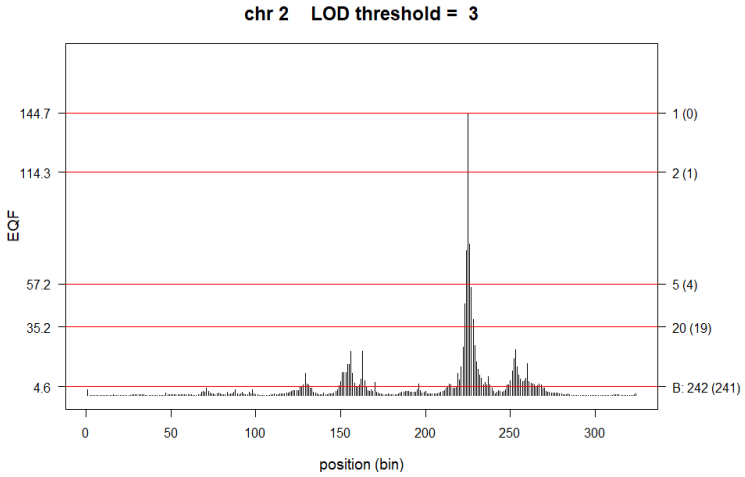

Figure 5: The EQF architecture of the 2^{nd} chromosome by using the EQF.plot() function. The EQF architecture are constructed by the uniform method with bin size of 0.5 cM.

In the `EQF.plot()` function, the result argument is the input data list produced from the `LOD.QTLdetect()` or `EQF.permu()` function. The command of `plot.all = TRUE` is to draw the EQF architecture of the entire genome (16 chromosomes) is outlined in a single figure (Figure 4). If `plot.chr = TRUE`, it will draw the EQF architectures of genome separately by chromosomes in different figures. Figure 5 shows the EQF architecture of one of the 16 chromosomes (the 2^{nd} chromosome).

The GRAMENE rice database example

The `Qhot()` function manages summarized QTL data collected from public QTL databases to detect QTL hotspots. Below, we demonstrate the use of the `Qhot()` function to detect QTL hotspots in the public GRAMENE rice database. First the QTL data in the GRAMENE rice database can be retrieved from GitHub using the following command:

```
load(url("https://github.com/py-chung/QTLEMM/raw/main/inst/extdata/gramene.chr.RDATA"))
load(url("https://github.com/py-chung/QTLEMM/raw/main/inst/extdata/gramene.QTL.RDATA"))
head(gramene.chr)

#>   CHR Center.cM. Length.cM.
#> 1   1       74.2        184
#> 2   2       55.5        161
#> 3   3       84.3        166
```

```
#> 4   4       19.7        133
#> 5   5       51.8        121
#> 6   6       66.8        127

head(gramene.QTL)

#>   X       Trait chr     L     R
#> 1 1 Biochemical   1  54.1  54.1
#> 2 2       Vigor   1 147.4 147.4
#> 3 3       Vigor   1 147.4 147.4
#> 4 4       Vigor   1 147.4 147.4
#> 5 5       Vigor   1 147.4 158.6
#> 6 6       Vigor   1  54.1  54.1
```

The `gramene.chr` is a data frame containing the information about the chromosomes, including their numbers, midpoint positions (in cM), and lengths. The gramene.QTL is a data frame for the information about QTLs, including their serial numbers, trait names, the chromosomes on which they are located, and positions of their flanking markers (in cM). Then the `Qhot()` function can utilize the information about chromosomes and QTLs, `gramene.chr` and `gramene.QTL`, to detect the QTL hotspots and output the analysis results.

```
result <- Qhot(gramene.QTL, gramene.chr, save.pdf = T)
names(result)

#> [1] "EQF"         "P.threshold" "Q.threshold" "nHot"
```

The analysis results include the EQF values at every bin of chromosomes (`EQF`), EQF thresholds obtained by the Yang et al. method (`P.threshold`), EQF thresholds obtained using the Q method (`Q.threshold`), and the numbers of detected hotspots in each chromosome by the Yang et al. method and Q method (`nHot`). The `save.pdf = T` command is to generate a PDF file that contains the plots of QTL composition at every bin. Figure 6 shows the plot of QTL composition at bin [7,8) of the first chromosome. It outlines the EQF architecture of the 1^{st} chromosome, the QTL intervals, and the composition of QTLs responsible for different traits in the hotspot at bin [7,8). Please refer to Yang, Wu, and Kao (2019) and P.-Y. Wu, Yang, and Kao (2021) for more details.

5 Conclusion and Discussion

In this paper we introduce the R package called **QTLEMM** for QTL mapping and QTL hotspot detection, and attempt to provide a comprehensive overview of the functions in the package by analyzing the examples of both simulated and real data sets. The package offers several advantages:

- **QTLEMM** is designed to accommodate a wide range of experimental populations, including backcross, F2, advanced intercrossed, and recombinant inbred populations. This versatility enables comprehensive QTL mapping analysis across different genetic backgrounds and breeding designs.
- Users can employ single-QTL or multiple-QTL models to estimate QTL parameters. It can accommodate a host of statistical models to be fitted and compared for QTL detection. The thresholds for claiming the QTL detection can be also determined across various experimental populations.
- **QTLEMM** handles both complete genotyping and selective genotyping data in QTL mapping analysis.
- Results from QTL mapping and hotspot detection analyses are presented through numerical and graphical outputs, facilitating interpretation and visualization of findings.
- **QTLEMM** is unique in providing the asymptotic variance-covariance matrix for estimates of QTL parameters and computing LRT.

The process of QTL mapping and hotspot detection usually involves the analysis of a large number of positions along the genomes. At each position, statistical models are applied to the estimation and testing for making decision, causing the process often typically time-consuming and computationally intensive in the analysis. We attempt to reduce the computational cost and speed up the analysis by eliminating unnecessary loops in writing the functions of this package. The **QTLEMM** package offers mature, effective, and commonly used statistical methods for QTL mapping and hotspot detection in the analysis of genetic architecture of quantitative traits. We envision the **QTLEMM** package will be valuable for finding more significant results in exploring the networks among genes, QTL hotspots and quantitative traits in broad areas of biological studies.

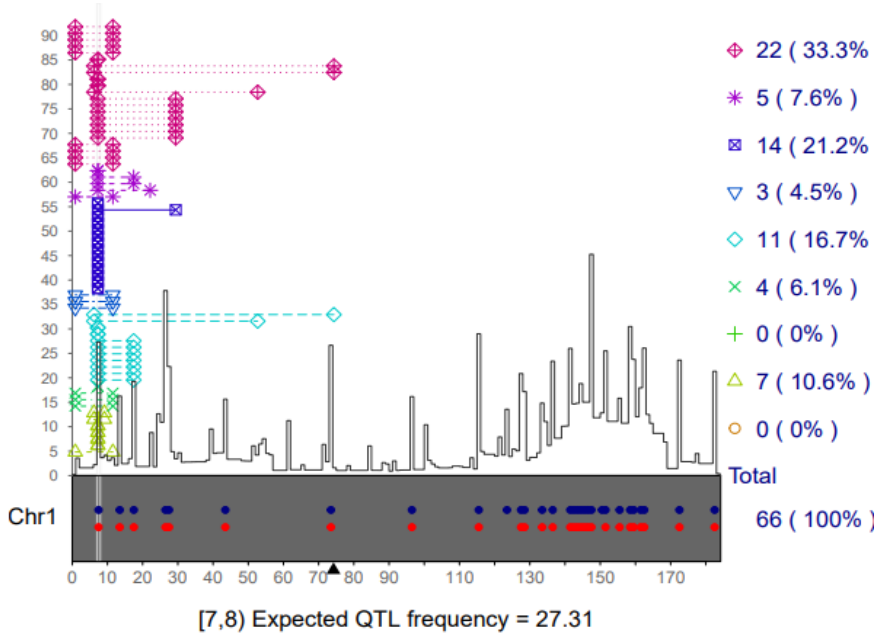

Figure 6: The plot of EQF architecture of the 1st chromosome and breakdown of QTL composition at bin [7,8) in the PDF file produced by using the Qhot() function with save.pdf = T command. The x-axis denotes the 1^{st} chromosome, the y-axis denotes the EQF values. The black triangle denotes the position of centromere. The blue (red) dots denote the QTL hotspots detected by the Yang, Wu, and Kao (2019) method (the Q method). The nine different colored symbols denote the QTLs responsible for the nine different trait categories (see Yang, Wu, and Kao 2019 for the nine different trait categories). The dotted lines denote the lengths of the marker intervals flanking the QTLs (QTL intervals). In total, 66 QTLs contribute probabilities to the EQF value of 27.31 at bin [7,8). The numbers of contributive QTLs of the nine different trait categories are 22, 5, 14, 3, 11, 4, 0, 7 and 0, respectively.

Availability

- The **QTLEMM** package is freely available from the Comprehensive R Archive Network at https://cran.r-project.org/web/packages/QTLEMM/index.html.
- The development website is available at https://github.com/py-chung/QTLEMM.

Acknowledgements

The research was partially supported by Ministry of Science and Technology of Taiwan under grants MOST 111-2118-M-001-005-. The authors declare that they have no conflict of interest.

References

Breitling, Rainer, Yang Li, Bruno M Tesson, Jingyuan Fu, Chunlei Wu, Tim Wiltshire, Alice Gerrits, et al. 2008. "Genetical Genomics: Spotlight on QTL Hotspots." *PLoS Genetics* 4 (10): e1000232. https://doi.org/10.1371/journal.pgen.1000232.

Brem, Rachel B, John D Storey, Jacqueline Whittle, and Leonid Kruglyak. 2005. "Genetic Interactions Between Polymorphisms That Affect Gene Expression in Yeast." *Nature* 436 (7051): 701–3. https://doi.org/10.1038/nature03865.

Broman, Karl W, Hao Wu, Śaunak Sen, and Gary A Churchill. 2003. "R/Qtl: QTL Mapping in Experimental Crosses." *Bioinformatics* 19 (7): 889–90. https://doi.org/10.1093/bioinformatics/btg112.

Chang, Myron N, Rongling Wu, Samuel S Wu, and George Casella. 2009. "Score Statistics for Mapping Quantitative Trait Loci." *Statistical Applications in Genetics and Molecular Biology* 8 (1). https://doi.org/10.2202/1544-6115.1386.

Chardon, Fabien, Bérangere Virlon, Laurence Moreau, Matthieu Falque, Johann Joets, Laurent Decousset, Alain Murigneux, and Alain Charcosset. 2004. "Genetic Architecture of Flowering Time in Maize as Inferred from Quantitative Trait Loci Meta-Analysis and Synteny Conservation with the

Rice Genome." *Genetics* 168 (4): 2169–85. https://doi.org/10.1534/genetics.104.032375.

Chen, Jing, Lindsey Leach, Jixuan Yang, Fengjun Zhang, Qin Tao, Zhenyu Dang, Yue Chen, and Zewei Luo. 2021. "A Tetrasomic Inheritance Model and Likelihood-Based Method for Mapping Quantitative Trait Loci in Autotetraploid Species." *New Phytologist* 230 (1): 387–98. https://doi.org/10.1111/nph.16413.

Churchill, Gary A, and RW1206241 Doerge. 1994. "Empirical Threshold Values for Quantitative Trait Mapping." *Genetics* 138 (3): 963–71. https://doi.org/10.1093/genetics/138.3.963.

Cockerham, C Clark. 1954. "An Extension of the Concept of Partitioning Hereditary Variance for Analysis of Covariances Among Relatives When Epistasis Is Present." *Genetics* 39 (6): 859. https://doi.org/10.1093/genetics/39.6.859.

Cox, David Roxbee, and David Victor Hinkley. 1979. *Theoretical Statistics*. CRC Press.

Dempster, Arthur P, Nan M Laird, and Donald B Rubin. 1977. "Maximum Likelihood from Incomplete Data via the EM Algorithm." *Journal of the Royal Statistical Society: Series B (Methodological)* 39 (1): 1–22. https://doi.org/10.1111/j.2517-6161.1977.tb01600.

Fu, Jingyuan, Joost JB Keurentjes, Harro Bouwmeester, Twan America, Francel WA Verstappen, Jane L Ward, Michael H Beale, et al. 2009. "System-Wide Molecular Evidence for Phenotypic Buffering in Arabidopsis." *Nature Genetics* 41 (2): 166–67. https://doi.org/10.1038/ng.308.

Geiringer, Hilda. 1944. "On the Probability Theory of Linkage in Mendelian Heredity." *The Annals of Mathematical Statistics* 15 (1): 25–57. https://doi.org/10.1214/aoms/1177731313.

Guo, You-Tsz. 2011. "A Study of Assessing Genome-Wise Statistical Significance of QTL Mapping in the Advanced Backcross Populations." Master's thesis, Taiwan, ROC: National Taiwan University.

Haldane, JBS, and CH Waddington. 1931. "Inbreeding and Linkage." *Genetics* 16 (4): 357. https://doi.org/10.1093/genetics/16.4.357.

Haldane, John BS. 1919. "The Combination of Linkage Values and the Calculation of Distances Between the Loci of Linked Factors." *J Genet* 8 (29): 299–309.

Haley, Chris S, and Sarah A Knott. 1992. "A Simple Regression Method for Mapping Quantitative Trait Loci in Line Crosses Using Flanking Markers." *Heredity* 69 (4): 315–24. https://doi.org/10.1038/hdy.1992.131.

Jansen, Ritsert C. 1993. "Interval Mapping of Multiple Quantitative Trait Loci." *Genetics* 135 (1): 205–11. https://doi.org/10.1093/genetics/135.1.205.

Kao, Chen-Hung. 2000. "On the Differences Between Maximum Likelihood and Regression Interval Mapping in the Analysis of Quantitative Trait Loci." *Genetics* 156 (2): 855–65. https://doi.org/10.1093/genetics/156.2.855.

———. 2004. "Multiple-Interval Mapping for Quantitative Trait Loci Controlling Endosperm Traits." *Genetics* 167 (4): 1987–2002. https://doi.org/10.1534/genetics.103.021642.

———. 2006. "Mapping Quantitative Trait Loci Using the Experimental Designs of Recombinant Inbred Populations." *Genetics* 174 (3): 1373–86. https://doi.org/10.1534/genetics.106.056416.

Kao, Chen-Hung, and Hsiang-An Ho. 2012. "A Score-Statistic Approach for Determining Threshold Values in QTL Mapping." *Front. Biosci. E* 4: 2670–82. https://doi.org/10.2741/e582.

Kao, Chen-Hung, and Miao-Hui Zeng. 2009. "A Study on the Mapping of Quantitative Trait Loci in Advanced Populations Derived from Two Inbred Lines." *Genetics Research* 91 (2): 85–99. https://doi.org/10.1017/S0016672309000081.

———. 2010. "An Investigation of the Power for Separating Closely Linked QTL in Experimental Populations." *Genetics Research* 92 (4): 283–94. https://doi.org/10.1017/S0016672310000273.

Kao, Chen-Hung, and Zhao-Bang Zeng. 1997. "General Formulas for Obtaining the MLEs and the Asymptotic Variance-Covariance Matrix in Mapping Quantitative Trait Loci When Using the EM Algorithm." *Biometrics* 53 (2): 653–65. https://doi.org/10.2307/2533965.

———. 2002. "Modeling Epistasis of Quantitative Trait Loci Using Cockerham's Model." *Genetics* 160 (3): 1243–61. https://doi.org/10.1093/genetics/160.3.1243.

Kao, Chen-Hung, Zhao-Bang Zeng, and Robert D Teasdale. 1999. "Multiple Interval Mapping for Quantitative Trait Loci." *Genetics* 152 (3): 1203–16. https://doi.org/10.1093/genetics/152.3.1203.

Kumar, Pardeep, Ningthai Longmei, Mukesh Choudhary, Mamta Gupta, Bhupender Kumar, BS Jat, Bharat Bhushan, Manesh Chander Dagla, and Sumit Kumar Aggarwal. 2024. "Enhancement of Nutritional Quality in Maize Grain Through QTL-Based Approach." *Cereal Research Communications* 52 (1): 39–55. https://doi.org/10.1007/s42976-023-00378-2.

Lander, Eric S, and David Botstein. 1989. "Mapping Mendelian Factors Underlying Quantitative Traits Using RFLP Linkage Maps." *Genetics* 121 (1): 185–99. https://doi.org/10.1093/genetics/121.1.185.

Lebowitz, RJ, M Soller, and JS Beckmann. 1987. "Trait-Based Analyses for the Detection of Linkage Between Marker Loci and Quantitative Trait Loci in Crosses Between Inbred Lines." *Theoretical and Applied Genetics* 73: 556–62. https://doi.org/10.1007/BF00289194.

Lee, Hsin-I, Hsiang-An Ho, and Chen-Hung Kao. 2014. "A New Simple Method for Improving QTL Mapping Under Selective Genotyping." *Genetics* 198 (4): 1685–98. https://doi.org/10.1534/

genetics.114.168385.

Li, Shaoyu, Qing Lu, and Yuehua Cui. 2010. "A Systems Biology Approach for Identifying Novel Pathway Regulators in eQTL Mapping." *Journal of Biopharmaceutical Statistics* 20 (2): 373–400. https://doi.org/10.1080/10543400903572803.

Mackay, Trudy FC, and Robert RH Anholt. 2024. "Pleiotropy, Epistasis and the Genetic Architecture of Quantitative Traits." *Nature Reviews Genetics*, 1–19. https://doi.org/10.1038/s41576-024-00711-3.

Meng, Qing-Lin, Cheng-Gen Qiang, Ji-Long Li, Mu-Fan Geng, Ning-Ning Ren, Zhe Cai, Mei-Xia Wang, et al. 2024. "Genetic Architecture of Ecological Divergence Between Oryza Rufipogon and Oryza Nivara." *Molecular Ecology*, e17268. https://doi.org/10.1111/mec.17268.

Neto, Elias Chaibub, Mark P Keller, Andrew F Broman, Alan D Attie, Ritsert C Jansen, Karl W Broman, and Brian S Yandell. 2012. "Quantile-Based Permutation Thresholds for Quantitative Trait Loci Hotspots." *Genetics* 191 (4): 1355–65. https://doi.org/10.1534/genetics.112.139451.

Piepho, Hans-Peter. 2001. "A Quick Method for Computing Approximate Thresholds for Quantitative Trait Loci Detection." *Genetics* 157 (1): 425–32. https://doi.org/10.1093/genetics/157.1.425.

Rebai, Ahmed, Bruno Goffinet, and Brigitte Mangin. 1994. "Approximate Thresholds of Interval Mapping Tests for QTL Detection." *Genetics* 138 (1): 235–40. https://doi.org/10.1093/genetics/138.1.235.

Sen, Śaunak, and Gary A Churchill. 2001. "A Statistical Framework for Quantitative Trait Mapping." *Genetics* 159 (1): 371–87. https://doi.org/10.1093/genetics/159.1.371.

Van Der Veen, JH. 1959. "Tests of Non-Allelic Interaction and Linkage for Quantitative Characters in Generations Derived from Two Diploid Pure Lines." *Genetica* 30: 201–32. https://doi.org/10.1007/BF01535675.

Vaughan, DA, E Balazs, and JS Heslop-Harrison. 2007. "From Crop Domestication to Super-Domestication." *Annals of Botany* 100 (5): 893–901. https://doi.org/10.1093/aob/mcm224.

Wang, Jia, Huihui Yu, Xiaoyu Weng, Weibo Xie, Caiguo Xu, Xianghua Li, Jinghua Xiao, and Qifa Zhang. 2014. "An Expression Quantitative Trait Loci-Guided Co-Expression Analysis for Constructing Regulatory Network Using a Rice Recombinant Inbred Line Population." *Journal of Experimental Botany* 65 (4): 1069–79. https://doi.org/10.1093/jxb/ert464.

Weir, BS, and C Clark Cockerham. 1977. *Two-Locus Theory in Quantitative Genetics*. https://www.cabidigitallibrary.org/doi/full/10.5555/19780134831.

West, Marilyn AL, Kyunga Kim, Daniel J Kliebenstein, Hans Van Leeuwen, Richard W Michelmore, RW Doerge, and Dina A St. Clair. 2007. "Global eQTL Mapping Reveals the Complex Genetic Architecture of Transcript-Level Variation in Arabidopsis." *Genetics* 175 (3): 1441–50. https://doi.org/10.1534/genetics.106.064972.

Wu, Chunlei, David L Delano, Nico Mitro, Stephen V Su, Jeff Janes, Phillip McClurg, Serge Batalov, et al. 2008. "Gene Set Enrichment in eQTL Data Identifies Novel Annotations and Pathway Regulators." *PLoS Genetics* 4 (5): e1000070. https://doi.org/10.1371/journal.pgen.1000070.

Wu, Po-Ya, Man-Hsia Yang, and Chen-Hung Kao. 2021. "A Statistical Framework for QTL Hotspot Detection." *G3* 11 (4): jkab056. https://doi.org/10.1093/g3journal/jkab056.

Xu, Shizhong, and William R Atchley. 1995. "A Random Model Approach to Interval Mapping of Quantitative Trait Loci." *Genetics* 141 (3): 1189–97. https://doi.org/10.1093/genetics/141.3.1189.

Xu, Shizhong, and Claus Vogl. 2000. "Maximum Likelihood Analysis of Quantitative Trait Loci Under Selective Genotyping." *Heredity* 84 (5): 525–37. https://doi.org/10.1046/j.1365-2540.2000.00653.

Yang, Man-Hsia, Dong-Hong Wu, and Chen-Hung Kao. 2019. "A Statistical Procedure for Genome-Wide Detection of QTL Hotspots Using Public Databases with Application to Rice." *G3: Genes, Genomes, Genetics* 9 (2): 439–52. https://doi.org/10.1534/g3.118.200922.

Zeng, Zhao-Bang. 1993. "Theoretical Basis for Separation of Multiple Linked Gene Effects in Mapping Quantitative Trait Loci." *Proceedings of the National Academy of Sciences* 90 (23): 10972–76. https://doi.org/10.1073/pnas.90.23.10972.

———. 1994. "Precision Mapping of Quantitative Trait Loci." *Genetics* 136 (4): 1457–68. https://doi.org/10.1093/genetics/136.4.1457.

Zeng, Zhao-Bang, Chen-Hung Kao, and Christopher J Basten. 1999. "Estimating the Genetic Architecture of Quantitative Traits." *Genetics Research* 74 (3): 279–89. https://doi.org/10.1017/S0016672399004255.

Zou, Guangyong. 2004. "A Modified Poisson Regression Approach to Prospective Studies with Binary Data." *American Journal of Epidemiology* 159 (7): 702–6. https://doi.org/10.1093/aje/kwh090.

Ping-Yuan Chung
Institute of Statistical Science, Academia Sinica
Taipei 11529, Taiwan, Republic of China
https://github.com/py-chung/QTLEMM

pychung@stat.sinica.edu.tw

You-Tsz Guo
Institute of Statistical Science, Academia Sinica
Taipei 11529, Taiwan, Republic of China
yowyow220@hotmail.com

Hsiang-An Ho
Institute of Statistical Science, Academia Sinica
Taipei 11529, Taiwan, Republic of China
r92221020@ntu.edu.tw

Hsin-I Lee
Institute of Statistical Science, Academia Sinica
Taipei 11529, Taiwan, Republic of China
hilee@stat.sinica.edu.tw

Po-Ya Wu
Institute of Statistical Science, Academia Sinica
Taipei 11529, Taiwan, Republic of China
debbywu@stat.sinica.edu,*tw*

Man-Hsia Yang
Crop Science Division, Taiwan Agricultural Research Institute, Council of Agriculture
Taichung 41362, Taiwan, Republic of China
ymh@tari.gov.tw

Miao-Hui Zeng
Institute of Statistical Science, Academia Sinica
Taipei 11529, Taiwan, Republic of China
miaomiao@stat.sinica.edu.tw

Chen-Hung Kao
Institute of Statistical Science, Academia Sinica
Taipei 11529, Taiwan, Republic of China
https://staff.stat.sinica.edu.tw/chkao/
chkao@stat.sinica.edu.tw