Advances and Applications in Bioinformatics and Chemistry

Dovepress
open access to scientific and medical research

Open Access Full Text Article

ORIGINAL RESEARCH

Efficient algorithms for multidimensional global optimization in genetic mapping of complex traits

This article was published in the following Dove Press journal:
Advances and Applications in Bioinformatics and Chemistry
27 October 2010
Number of times this article has been viewed

Kajsa Ljungberg[1]
Kateryna Mishchenko[2]
Sverker Holmgren[1]

[1]Division of Scientific Computing, Department of Information Technology, Uppsala University, Uppsala, Sweden; [2]Department of Mathematics and Physics, Mälardalen University College, Västerås, Sweden

Abstract: We present a two-phase strategy for optimizing a multidimensional, nonconvex function arising during genetic mapping of quantitative traits. Such traits are believed to be affected by multiple so called quantitative trait loci (QTL), and searching for d QTL results in a d-dimensional optimization problem with a large number of local optima. We combine the global algorithm DIRECT with a number of local optimization methods that accelerate the final convergence, and adapt the algorithms to problem-specific features. We also improve the evaluation of the QTL mapping objective function to enable exploitation of the smoothness properties of the optimization landscape. Our best two-phase method is demonstrated to be accurate in at least six dimensions and up to ten times faster than currently used QTL mapping algorithms.

Keywords: global optimization, QTL mapping, DIRECT

Introduction

Most traits of medical or economic importance are quantitative. Examples are agricultural crop yield, growth rate in farm animals and blood pressure and cholesterol levels in humans. These traits are generally believed to be governed by a complex interplay between multiple genetic factors and the environment. One method to locate the genetic regions underlying a quantitative trait is known as quantitative trait locus (QTL) mapping. A QTL is a DNA region (locus, pl. loci), harboring a gene or a regulatory element affecting a quantitative trait. In a standard QTL mapping study, genetic data (genotype data) from an experimental population is used as input to a statistical model of the measured trait (phenotype data). The model fit and significance tests are performed using numerical algorithms implemented in a QTL mapping software. QTL mapping methods were reviewed by Doerge.[1]

Finding the most likely positions of d QTL influencing a trait corresponds to minimization of a d-dimensional nonconvex objective function (the outer problem) which is defined by the QTL model fit (the inner problem). The numerical analysis framework governing the QTL mapping computations is that of a separable nonlinear least squares problem.[2–4] However, the QTL mapping problem has several special features that have to be accounted for, and standard optimization algorithms cannot be immediately applied. To derive and study efficient algorithms for the real world QTL mapping problems, we need to use a combination of knowledge from the fields of numerical analysis and genetics, and also rely on results from both numerical experiments and analysis for simplified problems.

In standard QTL mapping software,[5–8] the outer problem is solved using an exhaustive grid search. The computational requirement for this type of algorithm is

Correspondence: Sverker Holmgren
Division of Scientific Computing, Department of Information Technology, Uppsala University, Box 337, SE-751 05
Uppsala, Sweden
Tel +48 18 4712992
Fax +48 18 523049
Email sverker@it.uu.se

DOI: 10.2147/AABC.S9240

$\mathcal{O}(d^2G^d)$, where the number of grid points G is of the order 10^3. This type of scheme is reliable but prohibitively slow for $d > 2$, which has resulted in that high-dimensional searches have so far not been used in practice. In this paper, we introduce a hybrid global–local optimization algorithm for the outer problem, which is combined with an efficient scheme for solving the inner problem. Using the new algorithms, we show that it is possible to solve the optimization problems arising in QTL mapping up to at least $d = 6$ using a standard desk-top computer. We do not consider the important problem of how to select the QTL model, nor do we consider real application problems where high-dimensional QTL searches are performed for experimental data and genetic implications are drawn from the results. However, the introduction of the new algorithms paves the way for future work in these directions.

It should be noted that already today, geneticists routinely fit models with multiple QTL. This is performed using a forward selection procedure where an identified QTL is included as a known quantity when searching for an additional QTL. In this way it is possible to search for d QTL by a sequence of d one-dimensional exhaustive grid searches. For general QTL models, it is not clear how accurate this technique is. It could be anticipated that the forward selection scheme can fail to detect QTL that only affect the phenotype through interactions with other QTL. Several analyses of real data sets have revealed such interactions between pairs of QTL, some of which were only detectable by solving the full two-dimensional optimization problem.[9–11] Such results motivate our interest for developing efficient algorithms also for high-dimensional QTL mapping problems, and using simulated data we also show that the new scheme is more accurate than the forward selection technique for problems of this type.

A class of QTL models

A typical QTL mapping experiment involves two animal lines, each genetically homogeneous, of individuals that differ considerably in some phenotype. Genetic comparisons of the two groups would not reveal the QTL positions since the lines have a vast number of genetic differences, most of which are uncorrelated with the phenotype of interest. Instead, individuals are mated according to a specific scheme, most often the backcross or the intercross. This results in a population of offspring whose genetic material is a mosaic of DNA from the two original lines. The mosaic structure of the offspring genomes is the result of recombination, a random process which occurs during the formation of germ cells. Using standard (but still rapidly developing) experimental technology, the genotypes of each individual at a set of *marker loci* in the genome can be determined. The genetic markers are irregularly scattered at locations determined by the experimental procedure. Between markers, the genotypes can be estimated using a statistical model of the recombination process.

At each (initially unknown) QTL an individual may have only one of a few different genotypes. The QTL model describes how the phenotype depends on the individual's particular combination of genotypes at the QTL. Given a model including d QTL, the aim of QTL mapping is to find the set of d loci $xQTL$ where the genotype combinations best correlate with the phenotypic variation, and to determine whether the result is statistically significant. A robust approach for determining the significance thresholds is permutation testing,[12] where $\mathcal{O}(10^3)$ QTL searches are performed using randomly permuted data sets. If a model involving several QTL ($d > 1$) is used, this is of course a very computationally demanding procedure.

A standard class of models[13,14] for the phenotype of individual i, $i = 1,\ldots, n$, in the population is given by

$$y_i = \sum_{j=1}^{k_g} a_{ij}(x)b_j + \sum_{j=k_g+1}^{k_g+k_f} a_{ij}b_j + \varepsilon_i. \tag{1}$$

Here, y_i is the measured phenotypic value, k_g is the number of genotype parameters (in general modeling both QTL effects that are additive among loci, marginal effects, and nonlinear interaction effects between loci, epistatic effects), k_f is the number of covariates (or fixed effects), eg, sex and other known factors included in the model, a_{ij} are indicator variables for QTL genotypes and covariates, x is a set of d loci in the genome, b_j are the effects of the QTL and the covariates, and $\in_i$ is the error. The basis for the forward selection scheme mentioned in the Introduction is that, if a QTL has already been identified in a previous study, the genotype at that locus could be included in the model as a fixed effect.

In matrix form, the model Eq. (1) is given by

$$y = A(x)b + \varepsilon, \tag{2}$$

where y is the n-vector of observed phenotypes, A is the $n \times (k_g + k_f)$-matrix of indicator variables (the *design matrix*), b is the $(k_g + k_f)$-vector of effects, and $\in$ is an n-vector of errors.

The computational problem for QTL mapping

The inner problem, the linear regression method for QTL mapping will be introduced. Using this standard approach, any hypothetical set of d loci x directly corresponds to a matrix $A(x)$ which can be introduced in the model Eq. (2). The computational problem in QTL mapping is then to optimize the linear model fit over all possible positions x and to compute the corresponding residual sum of squares, RSS,

$$RSS = \min_{b,x} F(b,x) = \min_{b,x} \| A(x)b - y \|_2^2 . \tag{3}$$

The minimization problem of Eq. (3) arises in two versions; When searching for a putative set of QTL, the optimal set of loci x_{opt} and the corresponding RSS-value are needed (but the computation of the effects b_{opt} can normally be deferred until the significance of the result has been established). When performing the permutation test for determining the significance threshold, only good approximations of the RSS-values for the permuted problems are required.

The formula in Eq. (3) is a separable nonlinear least squares problem[2–4] where the model is given by a linear combination of nonlinear functions. Following,[2] the solution of Eq. (3) can be separated into two parts: The outer nonlinear problem,

$$\min_{x\in\mathcal{G}^d} f(x), \tag{4}$$

where the search space $\mathcal{G}^d$ is described later, and the inner linear problem, which has an explicit solution,

$$f(x) = \min_{b} \| A(x)b - y \|_2^2 = \| (A(x)\,A(x)^+ - I)\,y \|_2^2 , \tag{5}$$

where $A(x)^+$ is the pseudo-inverse of $A(x)$. If A has full rank, $A^+ = (A^TA)^{-1}A^T$.

The inner problem

At positions x where unambiguous genetic information is available, the matrix $A(x)$ in Eq. (2) is uniquely determined and $f(x)$ is given by Eq. (5). In practice, the genotypes are (at best) exactly known only at the marker loci. For a general x, exact genetic information is not available and the matrix entries $a_{ij}(x)$ are not given a priori. However, genetic recombination can be modeled as a Poisson process, and using the closest informative markers as input it is still possible to make a good estimate of the probability of a certain genotype. There are several different methods of forming the inner problem which all in some way use the Poisson process approach for handling the problem of missing data. For all these methods, the objective function reduces to Eq. (5) in the case of exact genotype information.

Interval mapping[13,15] gives maximum likelihood estimates of QTL locations and effects. In this case, the inner problem is nonlinear. The computations are expensive, since a nonlinear equation system must be solved for each solution of the inner problem. A commonly used alternative strategy is the linear regression method.[16–18] Here the genetic indicator variables are replaced by the a priori between-marker genotype probabilities given by the Poisson process model, and the corresponding design matrix is used in Eq. (5). Since the inner problem now is a single linear least squares problem (with some special features), this is a simple and fast method. When the quality of data is high (dense marker maps, few experimental errors) the global optimization landscapes and the QTL mapping results for interval mapping and the linear regression approximation are very similar.[17,19,20] For the experiments presented in this paper, we simulate such high-quality data and use the linear regression method. We use the same notation $A(x)$ and a_{ij} for both the exact and the approximated genotype variables. Given the matrix $A(x)$, effcient schemes for solving the least squares problem Eq. (5) in the QTL mapping setting are described by Ljungberg and colleagues.[21,22]

Efficient construction of the design matrix *A*(*x*)

Even when using the linear regression method, computing the matrix entries $a_{ij}(x)$ using the exact formulas for the a priori probabilities given by Haley and Knott[17] is rather costly. The closest informative markers need to be located, the genotype information retrieved, and then a set of exponential functions must be evaluated for each individual. Therefore, standard QTL mapping codes normally perform a preparation step by evaluating $a_{ij}(x)$ at regularly spaced grid points in the genome. Then, only values of $f(x)$ at points in this artificial grid are used in the exhaustive grid search employed for solving Eq. (4). The size of the genome is measured in the unit Morgan [M], which is a logarithmic function of the number of recombination events on an interval. A typical grid step size is $\wr(10^{-2})$ M and the size of a representative genome is $\mathcal{O}(10)$ M. The preparation step could be performed once and for all for each data set, and the resulting data stored to a file. However, since the objective function is only given at grid points, it is awkward to use optimization methods based on exploiting the piecewise continuity of the objective function (and its derivatives).

In this paper we use a new scheme for efficient evaluation of the matrix entries for the linear regression method at an arbitrary point in the search space. From the formulas described by Haley and Knott,[17] it is clear that the matrix entries $a_{ij}(x)$ can be completely described by a set of functions $g(\zeta)$, where ζ is a scalar variable that runs over marker intervals. For example, for a backcross population we have that, for each individual, the function $g(\zeta)$ between markers p and $p + 1$ is given by

$$g(\zeta) = K(1 + c_1e^{-2\zeta})(1 + c_2e^{2\zeta}) = K(1 + c_1c_2 + c_1e^{-2\zeta} + c_2e^{2\zeta}), \tag{6}$$

where $c_1 = \pm e^{-2k_1}$ is a constant where $k_1 \geq 0$ is the distance from marker p to the closest marker with known genotype to the left and the sign of c_1 depends on the individual's genotype at the informative marker, $c_2 = \pm e^{-2k_2}$ is a constant where $k_2 \geq D$ is the distance from marker p to the closest informative marker to the right and the sign of c_2 depends on the genotype at that informative marker. Note that genotype information might be missing at some markers. The parameter D is the distance between markers p and $p + 1$, and $K = 0.5/(1 + e^{-2(k_1+k_2)})$. For an intercross, two similar functions $g_1(\zeta)$ and $g_2(\zeta)$ are needed for each individual and interval, and for more general populations a few more functions of approximately the same form might be required.

To allow for efficient computation of the matrix entries, we approximate the functions $g(\zeta)$ by a corresponding set of third degree polynomials $p(\zeta)$.

The matrix entries should be continuous functions of x, implying that the four-parameter polynomials are fixed at the two markers and two degrees of freedom remain. We fit the polynomials

$$p(\zeta) = g(0)(1-\frac{\zeta}{D}) + \frac{\zeta}{D}g(D) + \zeta(\zeta - D)q_1 + \zeta^2(\zeta - D)q_2, \tag{7}$$

where q_1 and q_2 are the unknowns to be computed, by minimizing the integral of $g(\zeta)-p(\zeta)$ from 0 to D in the least squares sense. For typical values of D, third degree polynomials give sufficiently good approximations which do not suffer from oscillations. For the one-QTL backcross model, we have in fact proved that already using second degree polynomials the maximal error in the approximation $p(\zeta)$ is significantly smaller than the error arising from using a stepwise constant approximation corresponding to evaluating $g(\zeta)$ on a standard 1 cM grid. The proof, which is based on an interval analysis technique,[23] is rather extensive and is not shown here. If the marker interval is very short we fit only a one-degree polynomial to the endpoint data, and if the marker interval is unusually long we insert a pseudo-marker with no genotype information at the midpoint. The polynomial fitting is done in a preparation step. In the optimization algorithm, the matrix $A(x)$ needs to be computed for a given location x. In the implementation, the proper marker interval is first located and the corresponding polynomial coefficients are retrieved. Then the local coordinate ζ is determined and the polynomials are evaluated. Using this scheme, we can evaluate the matrix entries $a_{ij}(x)$ efficiently everywhere, not just at grid points. The extra time required for building the design matrix in this way, instead of retrieving grid values, is small compared to the computational time needed for an evaluation of the objective function. In total the increase in CPU time for the objective function is only about 10% compared with the grid-based strategy, and compared with using the exact functions this method is much faster. Finally, the memory requirement is significantly reduced compared to using a grid-based storage of $a_{ij}(x)$, since we only store a few parameters per marker interval.

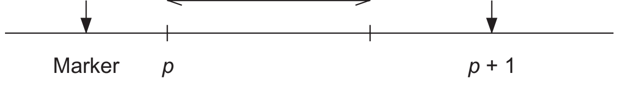

Figure 1. The parameters used for the polynomial approximation between markers p and $p + 1$.

The search space $\mathcal{G}^d$

The outer problem in Eq. (4) should in principle be solved by optimizing overall x in a d-dimensional hypercube where the side is given by the size of the genome. However, efficient optimization algorithms exploit more detailed information about the two-level structure of the search space $\mathcal{G}^d$. This structure also affects the properties of the solution to the inner problem, $f(x)$.

The first level of structure is given by that the genome is divided into C chromosomes, resulting in that the search space hypercube consists of a set of C^d d-dimensional unequally sized *chromosome combination boxes*, cc-boxes. A cc-box is identified by a vector of chromosome numbers $c = [c_1\ c_2 \ldots c_d]$, and consists of all x for which x_j is a point on chromosome c_j. The ordering of the loci does not affect $f(x)$. Therefore, we can restrict the search space g^d to cc-boxes identified by nondecreasing sequences of chromosomes. In addition, in cc-boxes where two or more edges span the same chromosome, eg, $c = [1\ 8\ 8]$, we need only consider values of x such that $x_k < (x_{k+1}-S)$ for k for which $c_k = c_{k+1}$. The distance S between two points on the same chromosome must be chosen large enough for some recombination to have occurred between x_k and x_{k+1}.

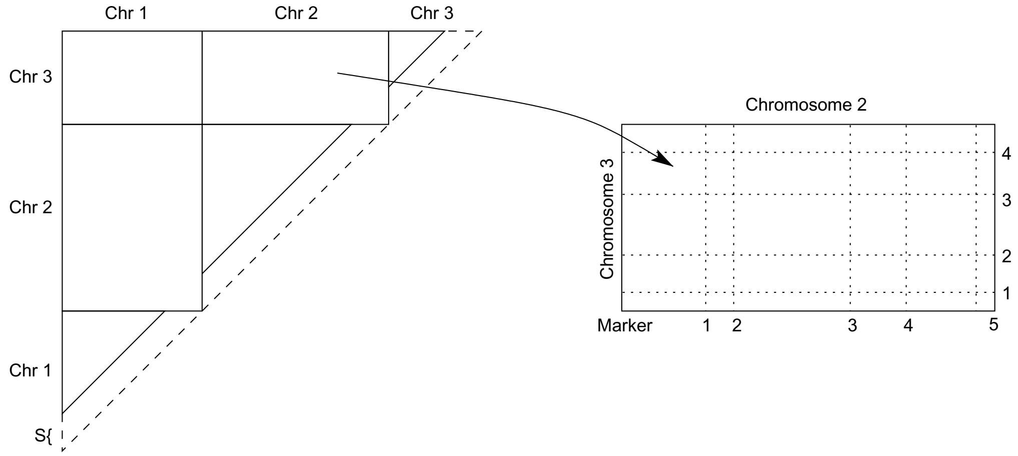

Figure 2 The search space for the outer problem is divided into chromosome combination boxes, cc-boxes. Each cc-box is further divided into marker boxes, m-boxes.

A second level of structure results from that, on each chromosome, the set of marker positions defines the locations where the genetic information is completely determined by the experimental procedure (in case of perfect data). Each cc-box is built up from a set of *d*-dimensional unequally sized marker boxes, m-boxes, defined by the marker positions and the endpoints of the chromosome. Figure 2 illustrates the two-level structure of the search space for a problem where $d = 2$. Figure 3 shows a part of a representative objective function $f(x)$.

We now make three observations concerning the properties of $f(x)$. Using Golub and Pereyra's theory for separable nonlinear least squares problem, more specifically the formulas for the variable projection functional and its gradient,[2] it would be possible to produce proofs of the statements below in a general setting. However, we instead focus on analyzing specific simple QTL mapping problems, using problem-specific properties in an attempt to try to retrieve more detailed information about the objective function:

(i) The function $f(x)$ is continuous within cc-boxes (but with discontinuities at chromosome boundaries) and smooth within m-boxes (but with discontinuous first derivatives at the m-box boundaries).

(ii) Within a cc-box, there exists a finite Lipschitz constant K for $f(x)$. For a simple case, ie, a one-QTL model for a backcross population, we have derived a tight

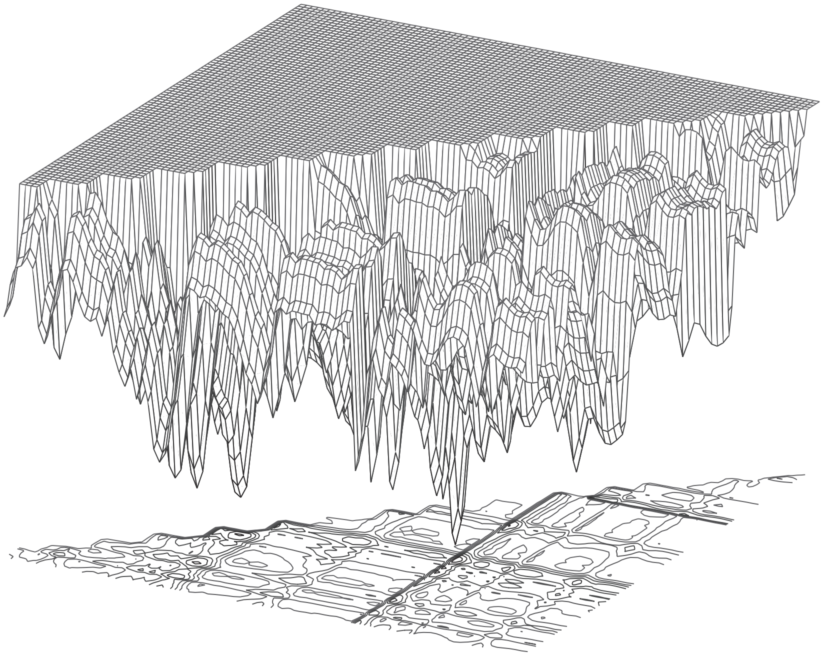

Figure 3 A part of a typical objective function $f(x)$ for the outer problem. The discontinuities at cc-box boundaries can be seen as straight lines in the contour plot below the surface.

bound for K. Again, we have chosen not to present the proof since it is extensive and the expression for K is complicated. Note that a good estimate of the Lipschitz constant could be used in the global optimization algorithm to discard regions of the search space. However, for the more complex models and high-dimensional searches which are of interest in this paper, it is much harder to derive corresponding estimates which are tight enough to be of practical value, and we have so far not pursued this approach further.

(iii) The function $f(x)$ is normally not convex within a cc-box, nor necessarily within an m-box. In fact, numerical experiments partly presented in Numerical experiments show that within an m-box, $f(x)$ may be convex or concave, or it may have none of these properties.

The outer problem

As exemplified by Figure 3, Eq. (4) is a global optimization problem with a large number of local minima. Apart from the standard exhaustive search, previously used optimization methods for QTL mapping problems include a genetic optimization algorithm, implemented for $d = 2$ using library routines,[24] and an algorithm based on the DIRECT[25] scheme, implemented for $d = 2$ and $d = 3$.[26] Ljungberg and colleagues,[26] results show that, for the problems considered, the DIRECT-based scheme is as reliable as an exhaustive grid search, and faster and more accurate than a genetic optimization algorithm. However, we have found that further improvements are needed to be able to tackle high-dimensional QTL mapping problems. The local minima of the objective function often have very similar function values, and for high-dimensional searches the DIRECT algorithm sometimes get stuck at the wrong local optimum for a long time. Also, once the correct local optimum is identified, the local convergence is still rather slow. A possible way of improving the convergence rate is to use a two-phase method, combining a global optimization algorithm with some more efficient algorithm for local optimization. A two-phase optimization algorithm for the outer problem we present methods of this type.

Before presenting the new algorithms, we give a brief review of DIRECT-based methods. The original scheme was presented as a general purpose deterministic global optimization algorithm for Lipschitz continuous multivariate functions subject to simple bounds.[25] In DIRECT the search space is divided into gradually smaller hyper-boxes, and the function value f_c is computed at the center of each box. If the Lipschitz constant K is known, a lower bound on the function value anywhere in a box can be computed as $f_c - K{\cdot}d_{cv}$, where d_{cv} is the center to vertex distance. This is the basis for Lipschitz optimization algorithms of branch-and-bound type for global optimization problems, see eg, Horst and colleagues' work[27] DIRECT does not require knowledge of the Lipschitz constant, but instead uses the approach that for a given K, the box with the lowest bound is potentially optimal and should be examined further. Jones and colleagues show how all boxes potentially optimal for *any* value of K can be identified, and each of these boxes is subdivided in a DIRECT iteration. Selecting boxes for subdivision amounts to determining the lower convex hull of the cloud of dots in a scatter plot of f_c versus d_{cv}, where each dot represents one box. Note that no regions in the search space are discarded by the algorithm, but the subdivision of "uninteresting" regions is postponed. DIRECT is sometimes referred to as a branch-without-bound algorithm. If a minimal box size is used and the algorithm is run sufficiently long, DIRECT will perform an exhaustive grid search on an equidistant grid defined by the centers of the minimal size boxes.

A number of variants of DIRECT have been described.[28–32] Several authors have noted that the final (local) convergence of the DIRECT algorithm often is rather slow. Nelson and Papalambros,[28] present an improved scheme where a quasi-Newton step is taken from the best current point and the box division pattern is adjusted accordingly. This is a theoretically attractive approach, but it is awkward to implement since the quasi-Newton steps break the simple box division pattern of the original algorithm and also may cross box boundaries. Cox and colleagues used[30] DIRECT to locate promising regions of the search space by running it until the smallest box reached a specified percentage of the original box size. Then a fixed number of the best points identified by DIRECT were used as starting points for local searches, using clustering to select only points which are sufficiently far apart from the others already used as starting points. A local optimizer based on sequential quadratic programming was used for the local optimization. The authors found that this version of DIRECT was suited for problems with many widely spaced local optima, which is a property also found in the QTL mapping problems. The DIRECT algorithm can also be made locally biased by using the infinity-norm instead of the euclidian norm when measuring d_{cv},[31] or globally biased by dividing each box which has the best f_c in its group of boxes of the same size, not only those on the convex hull.[33] The method described by Cox and colleagues[30] is an example of what Schoen[34] referred to as two-phase methods for global optimization. Here, the general strategy is to use one algorithm for global exploration of the search space and another for refinement of local optima, possibly iterating between the global and the local stages.

A two-phase optimization algorithm for the outer problem

Using the motivation earlier, we assume that the QTL mapping objective function $f(x)$ is Lipschitz-continuous within the cc-boxes. The arguments indicate that a Lipschitz-based algorithm could be a suitable choice for the outer problem (4). Since the objective function $f(x)$ is discontinuous at cc-box boundaries, the DIRECT algorithm described by Ljungberg and colleagues[26] is initiated by evaluating the objective function at the center of all cc-boxes. In this way, no assumption of continuity across cc-box boundaries is used. Also, symmetric cc-boxes ($c_k = c_{k+1}$ for at least one k) are taken care of by a special machinery in the box division algorithm. The multiple box initiation is a contrast to the original algorithm where there is only one hyper-box, spanning the whole space, on startup. Another difference is that the search space is not normalized to the unit hypercube.[26] This is to preserve the relation between the distance measure, Morgan, and change in the genotypes. Not normalizing the search space leads to that a large number of different values of d_{cv} emerge in the box selection step. In the original algorithm, this would lead to that too many boxes must be considered as candidates for division in each iteration. Ljungberg and colleagues[26] solved this problem by grouping boxes of similar size together using a hashing technique.

We now describe a two-phase algorithm using the DIRECT scheme described by Ljungberg and colleagues[26] for the global exploration of the search space. The basis of the algorithm is that when a box below a certain size (in max–norm) is chosen for subdivision in DIRECT, it is not divided according to the standard pattern. Instead, the box is first examined to determine whether it extends across one or more m-box boundaries. If this is the case, the box is divided into sub-boxes along these boundaries, resulting in in one (if no division is performed) or more boxes which all lie completely within an m-box. All boxes but the the one with the smallest function value at the center are returned to the global phase. A local algorithm chosen from the list below is used for optimization in the box with the smallest function value. According to the search space g^d, the objective function is smooth within this box implying that methods for local optimization using derivative information can be applied.

As for other practical global optimization methods, there is no theoretical well-founded convergence criterion for DIRECT. Jones and colleagues[25] suggested to terminate the search if no improvement of the objective has occurred during the last N_i iterations. In our two-phase algorithm we prefer to enforce a limit N_f on the number of function evaluations with no improvement. This form of stopping rule is easier to generalize to different data sets and different numbers of QTL d. Using this approach, we exploit information from the local optimization stages to determine how long to keep on performing DIRECT iterations. Note that this means that we in some cases do local optimization in many boxes, while sometimes only in one.

Local optimization methods

The standard method for solving separable nonlinear least squares problems is the variable projection algorithm, where the outer problem is solved using a Gauss–Newton method[2–4] which is adopted to the structure of the separated problem. For the QTL mapping problems, a global optimization method must first be employed to select the regions containing the most promising local optima. Moreover, once one or more such regions have been found, the methods of Gauss–Newton type reviewed[4] are not efficient for solving the local optimization problems. A Gauss–Newton method can be viewed as a scheme of quasi-Newton type, where the Hessian approximation is formed by neglecting a product where one of the terms is the residual.[35] The approach is viable if the residual is small and the residual functions are not highly nonlinear. However, the model is fitted to noisy biological data in the QTL mapping problems, and the residual is quite large even at the optimum. Instead, we have examined other local optimization methods.

In the numerical experiments presented, we compare the two-phase algorithms to the single-phase DIRECT algorithm presented by Ljungberg and colleagues.[26] For the local optimization stages in the two-phase algorithm we have employed the DIRECT, steepest descent, and quasi-Newton algorithms further described below. Using the polynomial approximations for the functions describing the matrix entries $a_{ij}(x)$ presented in Efficient construction of the design matrix $A(x)$, it would be possible to derive analytical formulas for the Jacobian and Hessian of the objective function. However, in practice this would be complicated already for the computation of the exact gradient because of the numerous variants of the model Eq. (1) that might occur. Therefore we have chosen to use numerical differentiation for computing the gradient.

- **DIRECT (D-D)**: A two-phase algorithm may of course use a global optimization algorithm also for the local stages. We use such an algorithm where we restart DIRECT in the box marked for local optimization. In our experiments, the local iteration is stopped when there is no function value improvement for the last two iterations. Note that this two-phase algorithm is not equivalent to a single global DIRECT run with more iterations.

- **Steepest Descent (D-SD)**: Using the steepest descent scheme, first-derivative information of the objective function is used in the most straightforward way. In this case, we stop the local search when the step length is smaller than a parameter γ, which is chosen as 1 cM in the experiments. During the line search along the negative gradient the Armijo condition is enforced, and the maximum step length is defined by the box boundary. The bound constraints are accounted for by a simplified barrier method, where a component of the negative gradient pointing out of the box is set to zero if the current point is close to a box boundary.
- **Quasi-Newton (D-QN)**: Using a quasi-Newton scheme, we also include approximative second derivative information in the local optimization algorithm. Here, we use the same line search and barrier method for the bound constraints as described for the steepest descent scheme, but choose the search direction using the BFGS method where an approximate inverse of the Hessian is repeatedly updated during the iterations using the gradients,[36] For the first iteration we perform a steepest descent step, and the approximative inverse Hessian is then initiated as a multiple of the identity matrix.[36] If the curvature condition is not satisfied, the inverse of the Hessian becomes indefinite. In such situations, we also perform a steepest descent iteration and restart the Hessian approximation process. The derivative across the boundary is discontinuous since a component of the gradient is set to zero as described above. However, the second derivative along an admissible search direction is continuous.

In the experiments, we compare the accuracy and efficiency of the two-phase algorithms described above to the one-phase DIRECT algorithm used in by Ljungberg and colleagues,[26] which we denote by **D**. Note that, for the **D** and **D-D** algorithms, we are guaranteed to get a sufficiently accurate approximation of the local minimum if the algorithms are run sufficiently long. For the **D-SD** and **D-QN** schemes, we are only guaranteed to reach the true minimum if the objective function is convex within the box where local optimization is applied.

Numerical experiments

We chose to test the new optimization algorithms on sets of simulated data. In this way, no explicit modeling errors are included in the computations (however, two types of noise *are* included), and also we know a priori approximately where the optimal position x_{QTL} is located. We have simulated a collection of 115 large data sets, imitating both backcross and intercross populations. The number of QTL, d, is varied from 2 to 6. In the intercross sets we only introduce marginal QTL effects, ie, effects depending only on the genotype at a single locus. In the backcross sets the major effects come exclusively from pairwise epistatic interactions, ie, they depend on the combined genotypes at pairs of loci. More details about the data are given in the Appendix.

We begin by presenting a numerical investigation of the properties of the objective function $f(x)$ in the search space hyper boxes where we perform local optimization. A simple midpoint test of a necessary but not sufficient condition for convexity of $f(x)$ was implemented in the line search algorithm within the **D-SD** and **D-QN** methods. If the condition was violated in any iteration for any line search in a hyper-box, that box was marked as nonconvex. The results of this investigation was that, of the boxes containing the global optima for the 115 test problems, in total 65 proved to be nonconvex. Further experiments indicated that in these cases, the function was concave and the minimum was located at the hyper-box boundary. The corresponding result for all boxes where local optimization was applied was that at least 43627 out of the 64739 boxes tested were nonconvex. From these results it would be tempting to draw the conclusion that the gradient-based optimization methods can not be used for the local optimization phase. However, we also performed an investigation of the accuracy of the different schemes. Here, the global optimum was considered to be found if $R < 1$, where R is the ratio of the current error to the accepted error, ie,

$$R = (f(x) – f(x_{opt}))/(f(x_{opt}) \cdot \gamma).$$

In the experiments we used $\gamma = 2 \cdot 10^{-4}$, a choice which is motivated by that the function value at the second smallest local minimum in some cases differ to the global minimum by almost as little as this. The slightly surprising result of the investigation was that all optimization methods, including the **D-SD** and **D-QN** schemes, succeeded in finding the global minimum for all the 115 data sets. We cannot give a rigorous explanation for the good result for the gradient-based methods.

Table 1 Stopping rule parameters

Algorithm	D	D-D	D-SD	D-QN
$p_{\text{alg}} \cdot G$	41	32	25	22

Before proceeding to a comparison of the work required for the different algorithms, we consider the criterion used for terminating the optimization algorithms again. As remarked in A two-phase optimization algorithm for the outer problem, we terminate the search for the global optimum when N_f objective function evaluations have been performed without any further improvement in the function value. For a d QTL model, the size of the search space is $G^d/d!$, where G is the length of the genome in centi-Morgan. This motivates us to set $N_f = (p_{\text{alg}} \cdot G)^{d}/d!$, where the parameters p_{alg} are determined by performing a large number of numerical experiments for each algorithm, adjusting p_{alg} so that the global optimum is found in all 115 data sets. In Table 1, the values of $p_{\text{alg}} \cdot G$ are shown. When performing the experiments resulting in Table 1, we noted that for all algorithms the values of p_{alg} were determined by a few "exceptional" data sets. For most sets of data, a much smaller value of p_{alg} can be used and the global optimum is still found. We also noted that, in general, the intercross data sets require more function evaluations than the backcross sets.

For QTL analysis problems, the evaluation of the objective function, ie, the solution of the inner problem, completely dominates the CPU time. In Tables 2 and 3, we compare the maximal number of objective function evaluations required for the different algorithms when solving all of the test problems. The tables show results for different values of d, and also include data for an exhaustive grid search with the resolution needed to locate the optima with the same accuracy as used for the other methods. In Figures 4 and 5, the same results are shown graphically using a logarithmic scale for the number of function evaluations.

From the tables and figures, it is clear that using a two-phase algorithm significantly reduces the number of function evaluations required, even when the DIRECT algorithm is used also for the local optimization. It is also clear that if the gradient based methods are employed, this gives a considerable further improvement. Here, the difference between the **D-SD** and **D-QN** schemes is not very large. It should also be noted that, as a result of the type of stopping criterion used, the number of function evaluations is dominated by the number of evaluations with no improvement required before terminating the global optimization algorithm. This also results in that there is no significant difference between the performance of the algorithms for the backcross and intercross data sets, even though for most of the backcross problems the global optimum is actually found faster than for most of the intercross problems.

In Tables 4 and 5, we show the fraction of the total number of objective function evaluations spent in the local phase for the three methods using a local algorithm. From the tables, it is clear that even though the difference in work between using DIRECT for the local optimization compared to using a gradient-based scheme is not dramatic, significantly less time is spent in the local optimization phases when the gradient-based methods are used. Also, for these schemes, the fraction of work in the local phase is less dependent on d.

Finally, we study the ability of the forward selection technique mentioned in the Introduction and A class of QTL models to locate the global optima for our test problems. We applied the forward selection procedure to our data sets, using exhaustive grid search for the consecutive one-dimensional optimization problems. In Table 6, we show the ratio of maximal actual error, in the minimum found using forward selection, to the accepted error. Recalling that a successful search is defined by $R \leq 1$, it is clear from the table that only for the model with $d = 2$ and the intercross data sets, the correct optimum is always found. For the backcross data, the method failed already for a model with two QTL. In many cases, the wrong cc-box is identified and the error is 50 times larger than acceptable. When the right cc-box is identified, the error in the position is often still very large. These results are consistent with previous observations for experimental data that forward selection can fail to detect

Table 2 The maximal number of function evaluations for different values of *d*, all back-cross data sets

d	2	3	4	5	6
Exhaustive grid search	13778625	$2 \cdot 10^{10}$	$3 \cdot 10^{13}$	$3 \cdot 10^{16}$	$3 \cdot 10^{19}$
D	2409	13501	126022	995586	>6650000
D-D	787	8571	59944	326606	1618725
D-SD	601	4296	25891	110433	418476
D-QN	530	3637	19980	71113	236636

Table 3 The maximal number of function evaluations for different values of *d*, all intercross data sets

d	2	3	4	5	6
Exhaustive grid search	13778625	$2 \cdot 10^{10}$	$3 \cdot 10^{13}$	$3 \cdot 10^{16}$	$3 \cdot 10^{19}$
D	1355	15010	124989	1341120	>6676400
D-D	1341	8985	54572	486633	1618411
D-SD	958	5445	24204	217926	415408
D-QN	778	4035	17310	149691	246884

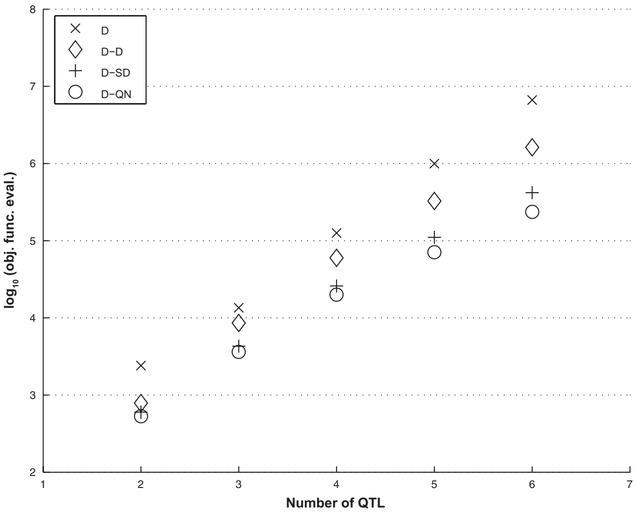

Figure 4 The maximal number of function evaluations for different values of d, all backcross data sets.

QTL whose main effect is epistatic.[9–11] It is important to note that also for the intercross data set, where there are no interaction effects at all, forward selection apparently can fail to find the correct cc-box. This indicates that it is important to use a true *d*-dimensional optimization method as soon as multiple QTL are fitted for a single phenotype, even when no interactions are included in the model. However, it is clear that this type of simple experiment needs to be extended to real data sets and actual QTL analysis problems before any firm conclusions of this type can be drawn.

Discussion

In this paper, we have discussed algorithms for QTL mapping using models including *d* QTL. Our approach is based on solving the full *d*-dimensional global optimization problem for determining the best model fit, which is in contrast to the traditional forward selection technique where a sequence of one-dimensional problems are solved.

Standard QTL mapping software uses an exhaustive search algorithm for solving the global optimization problem. For this type of algorithm, the computational requirement for problems where $d > 2$ is prohibitive. In this paper, we have shown that by exploiting the specific structure of the QTL mapping problem, it is possible to derive much more efficient algorithms. Using these schemes, the optimization problems for models with up to six QTL can be solved using a standard computer. For a ix-QTL problem, the best new algorithm is more than 10^{14} times more efficient than the standard exhaustive search would be.

The new algorithms are based on the DIRECT scheme for global optimization combined with different algorithms for local optimization in hyper-boxes which contain interesting objective function values. For the local optimization stages, both DIRECT and standard gradient-based schemes are examined. Since the objective function is often not convex within the hyper-boxes where local optimization is applied, it is not a priori clear that the gradient-based schemes will be able to correctly locate the global minima. However, numerical experiments for all 115 data sets show that this is indeed the case, and using these schemes results in an up to sevenfold increase in performance compared to a two-phase scheme where DIRECT is used both for global and local optimization. However, the latter algorithm is guaranteed to find the global minimum if run for sufficiently as many iterations, which is clearly not the case if a gradient-based method is used.

We also investigate the ability of the standard forward selection technique to locate the global optima for our test problems, and confirm the assumption that this approach can

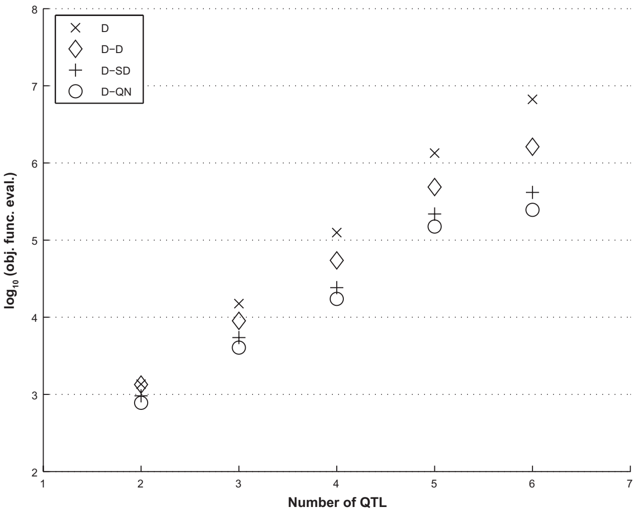

Figure 5 The maximal number of function evaluations for different values of *d*, all intercross data sets.

Table 4 The fraction of function evaluations in local algorithm, backcross data

d	2	3	4	5	6
D-D	0.49	0.75	0.80	0.87	0.89
D-SD	0.39	0.40	0.52	0.58	0.60
D-QN	0.31	0.40	0.48	0.52	0.52

Table 5 Fraction of function evaluations in local algorithm, intercross data

d	2	3	4	5	6
D-D	0.71	0.79	0.79	0.82	0.89
D-SD	0.56	0.66	0.51	0.61	0.60
D-QN	0.49	0.54	0.41	0.48	0.52

Table 6 Results from forward selection, ratio of actual error to accepted error

d	2	3	4	5	6
Backcross data	51, wrong cc-box	16	10, wrong cc-box	69, wrong cc-box	8
Intercross data	0.5	5	14	2, wrong cc-box	23

fail for models with several QTL. The conclusion is that, for our set of test problems, the new optimization methods are both more accurate and much more efficient that the methods currently used in QTL analysis software.

A suggested approach for using the new algorithms in practical QTL analysis is to exploit the two-phase DIRECT-DIRECT algorithm, using a strict stopping criterion, for determining x_{opt} and RSS_{opt} for the genetic data. The motivation for using DIRECT for the local search is that it is independent of the convexity properties of the objective function. Then the DIRECT-Quasi Newton scheme can be employed for optimization during the permutation test used for determining the significance of the result. In the significance testing, the effect of an eventual error in x_{opt} is not important.

The conclusion presented in this paper will have to be confirmed for real experimental data sets and real QTL analyses. This will also require some method for establishing a reference result for at least a few high-dimensional QTL mapping problems where the global optimum is not known a priori. One way of performing these extremely demanding computations is to use a parallel implementation of the exhaustive search algorithm.

Acknowledgments

Kajsa Ljungberg is affiliated with The Linnaeus Centre for Bioinformatics and The Graduate School in Mathematics and Computing, both at Uppsala University. The authors report no conflict of interest in this work.

References

1. Doerge R. Mapping and analysis of quantitative trait loci in experimental populations. *Nat Rev Genet*. 2002;3:43–52.
2. Golub G, Pereyra V. The differentiation of pseudo-inverses and nonlinear least squares problems whose variables separate. *SIAM J Numer Anal*. 1973;10:413–432.
3. Ruhe A, Wedin P. Algorithms for separable nonlinear least squares problems. *SIAM Rev*. 1980;22:318–337.
4. Golub G, Pereyra V. Separable nonlinear least squares: the variable projection method and its applications. *Inverse Probl*. 2003;19:1–26.
5. Lincoln S, Daly M, Lander E. *Mapping Genes Controlling Quantitative Traits with MAP-MAKER/QTL 1.1. Tech. Rep.* Whitehead Institute, 1992. Technical report. 2nd edition.
6. Basten C, Weir B, Zeng ZB. *QTL Cartographer, Version 1.15*. Raleigh, NC: Department of Statistics, North Carolina State University; 2001.
7. Seaton G, Haley C, Knott S, Kearsey M, Visscher P. QTL express: mapping quantitative trait loci in simple and complex pedigrees. *Bioinformatics*. 2002;18:339–340.
8. Broman K, Wu H, Sen S, Churchill G. R/qtl: QTL mapping in experimental crosses. *Bioinformatics*. 2003;19:889–890.
9. Shimomura K, Low-Zeddies S, King D. Genome-wide epistatic interaction analysis reveals complex genetic determinants of circadian behavior in mice. *Genome Res*. 2001;11:959–980.
10. Sugiyama F, Churchill G, Higgins D. Concordance of murine quantitative trait loci for salt-induced hypertension with rat and human loci. *Genomics*. 2001;71:70–77.
11. Carlborg Ö, Kerje S, Schütz K, Jacobsson L, Jensen P, Andersson L. A global search reveals epistatic interaction between QTL for early growth in the chicken. *Genome Res*. 2003;13:413–421.
12. Churchill G, Doerge R. Empirical threshold values for quantitative trait mapping. *Genetics*. 1994;138:963–971.
13. Soller M, Brody T, Genizi A. On the power of experimental design for the detection of linkage between marker loci and quantitative trait loci in crosses between inbred lines. *Theor Appl Genet*. 1976;47:35–39.
14. Moreno-Gonzalez J. Genetic models to estimate additive and non-additive effects of marker-associated QTL using multiple regression techniques. *Theor Appl Genet*. 1992;85:435–444.
15. Lander E, Botstein D. Mapping mendelian factors underlying quantitative traits using RFLP linkage maps. *Genetics*. 1989;121:185–199.
16. Knapp S, Bridges W, Birkes D. Mapping quantitative trait loci using molecular marker linkage maps. *Theor Appl Genet*. 1990;79:583–592.
17. Haley C, Knott S. A simple regression method for mapping quantitative trait loci in line crosses using flanking markers. *Heredity*. 1992;69:315–324.
18. Martinez O, Curnow R. Estimating the locations and the sizes of effects of quantitative trait loci using flanking markers. *Theor Appl Genet*. 1992;85:480–488.
19. Kao CH. On the differences between maximum likelihood and regression interval mapping in the analysis of quantitative trait loci. *Genetics*. 2000;156:855–865.
20. Sen S, Churchill G. A statistical framework for quantitative trait mapping. *Genetics*. 2001;159:371–387.
21. Ljungberg K, Holmgren S, Carlborg Ö. Efficient algorithms for quantitative trait loci mapping problems. *J Comput Bio*. 2002;9:793–804.
22. Ljungberg K. *Efficient Evaluation of the Residual Sum of Squares for Quantitative Trait Locus Models in the Case of Complete Marker Genotype Information. Tech. Rep. 2005-033*. Uppsala, Sweden: Division of Scientific Computing, Department of Information Technology, Uppsala University; 2005.
23. Moore R. *Interval Analysis.* Englewood Cliffs, NJ: Prentice Hall; 1966.
24. Carlborg Ö, Andersson L, Kinghorn B. The use of a genetic algorithm for simultaneous mapping of multiple interacting quantitative trait loci. *Genetics*. 2000;155:2003–2010.
25. Jones D, Perttunen C, Stuckman B. Lipschitzian optimization without the Lips-chitz constant. *Journal of Optimization Theory and Application*. 1993;79:157–181.
26. Ljungberg K, Holmgren S, Carlborg Ö. Simultaneous search for multiple QTL using the global optimization algorithm DIRECT. *Bioinformatics*. 2004;20:1887–1895.
27. Horst R, Pardalos P, Thoai N. *Introduction to Global Optimization, 2nd Edition*. Dordrecht, The Netherlands: Kluwer; 2000.
28. Nelson S, Papalambros P. A modification to Jones' global optimization algorithm for fast local convergence. In: *Proceedings of the 7th AIAA/NASA/USAF/ISSMO Symposium on Multidisciplinary Analysis and Optimization*. 1998 341–348.
29. Huyer W, Neumaier A. Global optimization by multilevel coordinate search. *Journal of Global Optimization*. 1999;14:331–355.
30. Cox S, Haftka R, Baker C, Grossman B, Mason W, Watson L. A comparison of global optimization methods for the design of a high-speed civil transport. *Journal of Global Optimization*. 2001;21:415–433.
31. Gablonsky J, Kelley C. A locally biased form of the DIRECT algorithm. *Journal of Global Optimization*. 2001;21:27–37.
32. Bartholomew-Biggs M, Parkhurst S, Wilson S. Using DIRECT to solve an aircraft routing problem. *Comput Optim Appl*. 2002;21:311–323.

33. Baker C, Watson L, Grossman B, Mason W, Haftka R. Parallel global aircraft configuration design space exploration. *International Journal of Computer Research*. 2001;10:no. 4.
34. Schoen F. Two-phase methods for global optimization. In: *Handbook of Global Optimization Volume 2*, ch. 5. Dordrecht, The Netherlands: Kluwer Academic Publishers; 2002.
35. Björck Å. *Numerical Methods for Least Squares Problems*. Philadelphia, PA: Society for Industrial and Applied Mathematics; 1996.
36. Nocedal J, Wright S. *Numerical Optimization*. New York, NY: Springer-Verlag, 1999.
37. Liu BH. *Statistical Genomics*. Baton Rouge, LA: CRC Press; 1998.

Appendix A

Data sets

The algorithms were tested on a collection of 115 simulated mouse data sets. They imitate the two most common experimental designs, the backcross and the intercross. Pseudomarker[20] was used to generate complete, dense auto-somal chromosome genotype information for a backcross with 1000 mice and an intercross with 500 mice. There were 92 markers in total, including one at the beginning and end of each of the 19 chromosomes. The average inter-marker distance was 15 cM and the standard deviation 7.6 cM. The complete genotype information at the set of markers, obtained from the simulation, was used as input in the preparation step for fast evaluation of the approximated a_{ij} (see Efficient construction of the design matrix $A(x)$).

Using the full genetic information at the markers ensures that the objective function values are fairly independent of the kernel method used. Sen and Churchill[20] demonstrated that linear regression and interval mapping kernels give very similar results and optimization landscapes as long as the proportion of missing genotype data is low.

The phenotypes were simulated according to the model Eq. (2). For each phenotype, QTL positions x_{QTL} and effects b_{QTL} were generated for a model with $d\ QTL$, $2 \leq d \leq 6$ (see below). $A(x_{QTL})$ was built, using exact genotype information from the Pseudomarker simulation, and $y_{QTL} = A(x_{QTL})$ b_{QTL} was computed. The noise vector $\in$ was constructed as the sum of two components, $\in = \in_{gen} + \in_{rand}$. The genotype dependent component $\in_{gen}$ was simulated by generating positions $x\in$ and effects v_j (see below), building $E(x_\in)$ in the same way as $A(x_{QTL})$ and computing $\in_{gen} = E(x_\in)v$. The random noise component $\in_{rand}$ was generated from a normal distribution with zero mean and variance $\sigma^2_{\in,rand}$. The variance was chosen to give the desired broad sense heritability H, given in Table A1. Here, the heritability is defined[37] as

$$H = \frac{\sigma^2_{QTL}}{\sigma^2_\in + \sigma^2_{QTL}}, \qquad \text{(A1)}$$

where σ^2_{QTL} is the variance of y_{QTL} and $\sigma^2_\in = \sigma^2_{\in,gen} + \sigma^2_{\in,rand}$ is the variance of the noise $\in$. The simulated phenotype is $y = yQTL + \in$.

Table A1 Heritabilities

d	2	3	4	5	6
H backcross	0.05	0.17	0.20	0.31	0.34
H intercross	0.08	0.15	0.20	0.27	0.34

If the heritability is too low, ie, the noise levels too high, the global optimum x_{opt} will be in a different cc-box than x_{QTL} and the true QTL locations undetectable. We simulate phenotypes with very low heritability, tuned so that x_{opt} is in the same cc-box as x_{QTL} but $f(x_{opt})$ is only slightly smaller than function values in other cc-boxes. This ensures that our test cases are difficult enough to be realistic.

Given the number of QTL d, the vector x_{QTL} was generated by randomly selecting d of the 19 autosomal mouse chromosomes and on each selected chromosome randomly place a QTL. Both the chromosome selection and the QTL placement was done using uniform probability distributions. The vector $x_\in$, always of length 10, was generated in the same way as x_{QTL}, except that only chromosomes not already harboring a QTL could be selected.

Epistatic interactions were simulated for the backcross data only. Interacting QTL pairs were formed by randomly grouping the previously placed d QTL into pairs. For even d, each QTL is in exactly one pair, while for odd d one QTL is part of two pairs.

Marginal QTL effects depend only on the genotype at a single locus. In an intercross the genotype at QTL x_k is described by two indicator variables, a_k^α and a_k^δ, and the phenotype effect of QTL k is $a_k^\alpha \cdot b_k^\alpha + a_k^\delta \cdot b_k^\delta$. In a backcross the genotype at QTL x_k is described by a single indicator variable, a_k^α, and the phenotype effect of QTL k is $a_k^\alpha \cdot b_k^\alpha$. Epistatic effects depend on the genotypes at two or more loci. An additive by additive pairwise interaction depends on the genotypes at two loci k and l, and the effect is $a_{kl}^{\alpha\times\alpha} \cdot b_{kl}^{\alpha\times\alpha}$ where $a_{kl}^{\alpha\times\alpha} = a_k^\alpha \cdot a_l^\alpha$.

The magnitude of the parameters b_k^α and $b_{kl}^{\alpha\times\alpha}$ were generated randomly from a uniform distribution $U(\mu - 0.2 \cdot \mu, \mu + 0.2 \cdot \mu)$, and the sign of the effects positive or negative with equal probability. The magnitude of the dominance effect of QTL k, where applicable, was chosen as $|b_k^\delta| = |(|b_k^\alpha| - \mu)|$ and the sign of the effect positive or negative with equal probability.

The means of the effects were defined relative to the intercross mean $\mu = \mu_{IC}$. (The fixed heritability makes the absolute level irrelevant.) In the backcross case, $\mu = 0.2 \cdot \mu_{IC}$ was used for marginal effects and $\mu = \mu_{IC}$ for interaction effects. The effects of the genotype dependent noise were generated exactly as the QTL effects, with $\mu_\in = 0.2 \cdot \mu_{IC}$.