Efficient evaluation of the residual sum of squares for least-squares problems in genetic mapping of complex traits

Kajsa Harling [a], Carl Nettelblad [a,*], Carl Nettelblad [a]

[a] *Uppsala University, Department of Information Technology, Division of Scientific Computing, Box 337, SE-751 05 Uppsala, Sweden*

Abstract

A core computation of many popular quantitative trait locus (QTL) methods for genetic mapping of complex traits is determining the residual sum of squares, RSS, for a regression of trait values on (pseudo-)marker genotypes. A single evaluation is easily performed using standard QR factorization, but together the RSS computations take considerable time and often constitute the major part of the computational effort. The presented algorithm for RSS evaluation is mathematically equivalent to evaluation via QR factorization but 10-100 times faster depending on the model and number of data dimensions. It can be used for all standard QTL models. The reduced computational complexity opens the possibility for more detailed data analysis and more extensive model comparisons in genetic mapping of complex traits.

Key words: Genetic analysis, QTL, least-squares, RSS

Introduction

The field of genetics has seen a remarkable revival during the last twenty years due to the development of highly efficient high-throughput methods for retrieving (and modifying) genetically encoded information. However, these developments need to be matched by development in the modulus and computational tools used for analysis of these vast amounts of information. One type of analysis is searching for *quantitative trait loci (QTL)*, i.e. genetic positions (loci) that are affecting a specific measurable trait, also called a phenotype.

* Corresponding author.
Email address: Carl.Nettelblad@it.uu.se (Carl Nettelblad).

Most complex traits like body weight, susceptibility to diseases like diabetes, fat content in meat, crop yield for rice etc. are affected both by genetic factors, QTLs, and the environment. Also, it can be expected that several QTL interact; the allele (gene version) in one locus will not have a predictable genetic effect unless the alleles in other loci are also known. In genetics, such interacting effects are referred to as *epistasis*. The result is that an appropriate QTL model will include both environmental covariates (including also fully known genetic factors like the sex of the individuals) and a multi-dimensional structure to account for allele variations in several loci. *QTL mapping* refers to the process of finding such sets of loci, starting from the experimentally determined genetic compositions of the individuals at a set of marker loci, using a statistical model and computational techniques to relate the genotypes to the phenotype studied.

Established methods for QTL mapping include interval mapping [23] and multiple interval mapping [21]. Reviews of statistical methods and computational challenges for QTL mapping in experimental crosses are given in [13] and [7]. Despite rapid development in computer hardware and algorithms, computational demand restricts the use of true multiple QTL methods. In particular, estimating the number of QTL and identifying their interactions is an issue where the computational complexity limits the space of models that can be investigated, see e.g. [9,3,8,13,31,36]. By improving the efficiency of the core computations of a statistical method, the same results can be obtained in less time, allowing for a more thorough investigation of the model space. Many statistical methods, although theoretically very different, share the computational sub-problems where trait values are regressed on marker genotypes. Some, including traditional variance analysis, only involve individuals with available genotypes, see e.g. [32,33,8,6]. Other authors propose multiple imputation for generating complete data realizations [3,30], or replacing uncertain observations with a set of weighted complete observations [20]. For all these methods, computing the residual sum of squares, RSS, for a possibly weighted least squares problem normally constitutes a major part of the computational effort.

Although this approach has not generally been used for QTL mapping, the RSS can be computed using matrix factorizations leading to expressions equivalent to simple functions of sums of squares and cross-products recognized from variance analysis, see e.g. [15]. The expressions are very fast to evaluate compared with computing a QR factorization of the design matrix, which is the standard least squares problem solution method. For complicated models in QTL analysis the expressions become more involved, but in this paper we show that a general algorithm that can accommodate epistasis as well as discrete and continuous covariates can be defined. We derive the PERF (*Pseudomarker Evaluation of RSS Function*) algorithm, which provides a new method for evaluating the RSS in the case of complete, possibly imputed, (pseudo-)marker

genotype information. As is shown in experiments, the gain in computing time is more than 10-fold compared with QR factorization already for small models, and up to 100-fold for larger models.

The problem of repeatedly evaluating QTL models is a specific case of a more general problem, the computation of RSS for factorial experiments. The general formulation of this problem has been studied for a long time and by several authors. Important contributions are presented in e.g. [35,17,12,14]. The PERF method is related to these well-known schemes, but it also exploits the structure of the QTL mapping problems, arriving at a highly efficient scheme for these specific but important problems.

The QTL mapping problem

The total genetic information of an individual, the genome, is distributed over a set of chromosomes. Focusing on the main body of the genome, i.e the autosomal (non-sex) chromosomes, every genetic location comes in two copies (originating from the two parents of the individual). Such a location is called a locus. In each locus, the possible variations are called alleles. The two copies can share the same allele, resulting in the individual being homozygous in that locus. If the two alleles are different, the individual is heterozygous instead. In a natural population, there can be multiple possible alleles in any locus, all of which might not even be known. In an experimental population, though, the variability can be assumed to be restricted to a limited number, e.g. 2. These alleles can be identified by a single binary number. The genotype (genetic state), in a locus can then described by a scalar c_i for each individual i, where $c_i \in \{0, 1, 2\}$, corresponding to zero, one or two alleles of e.g. the 0 type. Then, c_i is mapped to a binary row vector Z_i indicating the current out of three possible genetic states, resulting in the mapping

$$\begin{aligned} c_i = 0 \qquad & Z_i = \begin{pmatrix} 1 & 0 & 0 \end{pmatrix} \\ c_i = 1 \Leftrightarrow \quad & Z_i = \begin{pmatrix} 0 & 1 & 0 \end{pmatrix} \\ c_i = 2 \qquad & Z_i = \begin{pmatrix} 0 & 0 & 1 \end{pmatrix}. \end{aligned} \tag{1}$$

The true genotype will always be one of these three states. An indicator matrix $\mathbf{Z}$ for the genotype of a population of n individuals is formed by the

concatenation of the individual Z_i,

$$\mathbf{Z} = \begin{pmatrix} Z_1 \\ Z_2 \\ \vdots \\ Z_n \end{pmatrix} \tag{2}$$

In practice, genetic information is only available at some marker positions, i.e. the specific loci that are probed by the experiment that determines the genetic composition. Markers can be missing for some individuals due to experimental problems. Markers can also lack in variability, resulting in an inconclusive allele determination. These factors mean that the genotype information might not be conclusive for a specific locus, even if that locus coincides with a marker. In such cases, c_i becomes a discrete distribution rather than a scalar, with a probability mass function defined with non-negative values for $\{0, 1, 2\}$. This probability mass can be described by modifying Z_i to be non-binary vectors, the column in element j directly reflecting $p(c_i = j)$, while maintaining the condition

$$\forall i \sum_j Z_{ij} = 1. \tag{3}$$

This approach of probability-based indicator vectors is consistent with the genetic model description given in [18]. When analyzing multiple loci, the indicator matrix will be a Kronecker product of the corresponding matrices for the loci involved, assuming the genotype probabilities to be independent. Efficient Markov models can be used to determine genotype probabilities, even for non-independent loci, from marker data [29,19].

The full indicator matrix $\mathbf{Z}$ forms the mathematical representation of the genotype in a specific locus (or set of loci for a multi-locus model). The purpose of a QTL scan is to find the locus that best explains the phenotypic variation seen, interpreted as resulting in the minimal residual in a linear regression. This minimization problem can be expressed as

$$\arg\min_{\bar{x},\mathbf{E}} \left(M(\mathbf{Z}(\bar{x}))\mathbf{E} - y\right)^T \mathbf{W} \left(M(\mathbf{Z}(\bar{x}))\mathbf{E} - y\right) \tag{4}$$

where $\bar{x}$ is a vector of d loci in the genome, $\mathbf{Z}(\bar{x})$ is the indicator matrix for those loci, M is a transformation turning the indicator matrix into a (model) design matrix, $\mathbf{E}$ is a vector of genetic effects for the parametrization resulting from M, $\mathbf{W}$ is a (diagonal) weighting matrix and y is a vector of observed (scalar) phenotype values, with one row per individual. $\mathbf{W}$ would be the identity matrix I in an ordinary least-squares approach. M can also be chosen to be the identity, although one will frequently use other parametrizations,

and also possibly include non-genetic covariates representing environmental factors. One common structure is

$$M(\mathbf{Z}) = \mathbf{ZS} = \mathbf{A} \tag{5}$$

where $\mathbf{S}$ is considered to be a data-independent statistical formulation of the model of genetic effects [1], while $\mathbf{A}$ is the resulting composite regression matrix. For a given locus, the so-called F_2 model will formulate the genetic effects as a mean effect, one variable (a, additive effect) depending directly on allele count and one (d, dominant effect) describing the deviation from an additive model:

$$\mathbf{S}_{F_2} = \begin{pmatrix} 1 & -1 & -0.5 \\ 1 & 0 & 0.5 \\ 1 & 1 & -0.5 \end{pmatrix} \tag{6}$$

The minimization problem presented above is frequently solved as one scan/search part over putative $\bar{x}$, and a direct determination of the minimum RSS through conventional methods, such as the method of QR factorization of $W^{1/2}A$, see e.g. [5]. In some more elaborate methods, multiple model realizations are tested in each location, but the problem of each RSS computation is still identical. The details of the multi-dimensional global optimization problem has been investigated in for example [9,25] and is not directly relevant when evaluating the RSS in a specific position. For these reasons, the dependency on $\bar{x}$ can be dropped from all expressions, instead assuming an indicator matrix $\mathbf{Z}$ to is already given.

In this approach, the actual determination of the minimizing value of $\mathbf{E}$ can be deferred until the proper model has been selected, as suggested in [8]. In our presentation, the model parameterization of [28] is used, but it is important to note that the RSS will be equal using other models, e.g. from [11], where the parameters are linear combinations of the model of [28]. As will be shown below, an alternative formulation of Equation 4 leads to a highly efficient way of computing the RSS, substantially faster than QR factorization.

Recall that d is the number of QTL in the model. For any $\bar{x}$, the individuals will, depending on their genotypes at $\bar{x}$, belong to one of $c = g^d$ classes, where g is the number of possible genotypes. Including a discrete covariate for e.g. 4 different family effects can be seen as adding an extra locus with 4 possible genotypes, giving $c = g^d \cdot 4$. The number of individuals in class l is denoted by n_l, the sum of elements in a vector v for individuals in a class is denoted by $\Sigma_l v$, and the class phenotype mean is denoted μ_l. The Hadamard (element-wise) product of two vectors is indicated by $\odot$.

In the case of a full model, i.e. a model including all main effects and interac-

tions, the number of parameters equals the number of classes, and therefore the class phenotype means can be estimated independently. The standard variance analysis RSS formula is then

$$RSS = \sum_{l=1}^{c} SS_{yy,l} \tag{7}$$

where

$$SS_{yy,l} = \Sigma_l (y - \mu_l) \odot (y - \mu_l) = \Sigma_l (y \odot y) - \frac{(\Sigma_l y)^2}{n_l} \tag{8}$$

in the case of no weighting, which is the sum of squared errors for group l. The basis for this paper is that expressions of this type, simple functions of sums of squares and also sums of cross-products, are very fast to evaluate. In the case of non-full models the expressions become more complicated, making it awkward to derive formulas 'by hand'. Below, it is described how the algorithm to obtain general expressions can be easily implemented. Discrete and continuous covariates are easily included. The algorithm PERF has the computational efficiency of Equation 7 and is completely general.

A necessary condition for the efficient algorithms to be applicable is that the individuals can be sorted into genotype classes. The simplest example is when only the marker loci are considered in a search for single or multiple QTL and only genotyped individuals are included in the analysis. This strategy is becoming more realistic as experimental techniques and the density of marker sets improve. Another possibility is multiple imputation [30,3]. Then a set of complete (pseudo-)marker genotype information sets, consistent with the known data, is generated. The RSS is computed according to Equation 4, either for each data realization separately [30] or for the combined data set using an augmented design matrix $\mathbf{A}_{aug}$ with complete genotype information and giving each of the n_{imp} imputed observations the weight $1/n_{imp}$ [3]. The latter approach is, from an algorithmic viewpoint, related to a third method, namely the generalized linear finite mixture model method presented in [20], which consists of iterative refittings of such models. Individuals with uncertain genotypes are represented by several rows in the design matrix, and the rows of the resulting $\mathbf{A}_{aug}$ are weighted according to the probability of each genotype. In contrast to the method of [3], all possible genotypes are always included, each possibility is included exactly once, and the weights are iteratively refined.

It should be noted that the linear regression approximation to interval mapping [22,27,18] do in general not involve a least squares problem where individuals can be grouped into genotype classes, and are therefore not suited for the methods presented here. However, due to the loss of model orthogonality in those methods, one might choose to use an imputation approach for those as well.

The PERF method

To exploit the advantages of complete genotype information an alternative formulation to Equation 4 is used,

$$\begin{aligned} RSS &= y^T W y - y^T W A (A^T W A)^{-1} A^T W y \\ &= y^T W y - y^T W U P (P^T U^T W U P)^{-1} P^T U^T W y \\ &= y^T W y - y^T W U (L D L^T)^{-1} U^T W y \\ &= y^T W y - y^T W U L^{-T} D^{-1} L^{-1} U^T W y \\ &= y^T W y - z D^{-1} z \end{aligned} \tag{9}$$

where $U \in \mathrm{R}^{n \times k}$, $P \in \mathrm{R}^{k \times k}$, rank$(P)= k$, $U^T W U = L D L^T$ is the factorization of $U^T W U$ into a unit lower triangular matrix L, a diagonal matrix D, and the transpose of L, and z is the solution vector to the unit triangular system $Lz = U^T W y$. Starting with either the [11] or [28] model the same matrix U can be obtained by using different matrices P.

The computationally most expensive step when implementing Equation 9 is performing the matrix-matrix and matrix-vector multiplications to form $U^T W U$ and $U^T W y$. Once these terms are available the remaining computations are comparatively fast. For any model it is possible to choose an appropriate U that makes it easy to build $U^T W y$ and $U^T W U$ from the genotype class counts and phenotype sums. The use of a method that avoids performing the costly matrix multiplications gives a dramatic reduction in computing time, compared to obtaining the RSS via QR factorization or solution of the normal equations in the traditional way. Additional time savings are obtained by exploiting the sparsity pattern of the matrix $U^T W U$ during the $L D L^T$ factorization, and of the equation system $Lz = U^T W y$ when solving for z. Below, the matrix expressions that are used in the implementation of PERF, are presented, together with expressions based on sums of squares and cross-products. The latter are given for illustration only, in order to demonstrate the connection between the matrix elements and common statistical expressions. The software implementation uses matrix algebra only.

Some additional notation is needed for the presentation. Let each of the c genotype classes be identified with a d-digit code enclosed by square brackets, where the jth digit denotes the genotype at the jth pseudomarker, for example [13] or [22] when $d = 2$ and $g = 3$. The discrete covariate code is enclosed by angular brackets. Examples with continuous covariates, which do not change the number of classes, are shown below. The number of individuals in a class and phenotype sums are identified with the bracketed codes as subscripts, and an asterisk $*$ indicates a sum over all categories at that position. e.g. $n_{[*2]} = n_{[12]} + n_{[22]} + \ldots + n_{[g2]}$. When classes are identified by an index l, $1 \leq l \leq c$, no brackets are used.

For length reasons, only a few explicit examples of Equation 9 are given here. A comprehensive list of matrices for common QTL models can be found in the Appendix. The examples illustrate the fact that once the n_l and $\Sigma_l y$ terms are available, RSS can be obtained essentially for free for any model with the same set of classes. This is an important feature of the PERF method, since it is frequently relevant to try many different models when analyzing a single dataset, see e.g [8]. The incremental structure of the algorithm also makes it easy to quantify the reduction in RSS for each additional model parameter.

In the examples it is assumed that the matrix $W^{1/2}A$ has full rank. If it is rank deficient the RSS can still be computed but the model is inappropriate and the RSS should not be used. Detection of rank-deficiency is discussed in the Implementation section. The examples all assume $W = I$, i.e. all weights are equal to 1. For a general $W = diag(w_1, w_2, \ldots, w_n), w_i > 0$, the genotype class counts n_l in the formulas are replaced by the sums of weights $\Sigma_l w$, $y^T y$ by $y^T W y$ and the phenotype sums $\Sigma_l y$ by weighted sums $\Sigma_l (W y)$.

Full model: One special case is the full model, where the number of parameters is equal to the number of classes. Then, given that there is at least one individual in each genotype class, it is possible to pick $c = k$ unique rows from A to form a sub-matrix $S \in \mathrm{R}^{k \times k}$. Every genotype class is represented by one of these unique rows. If there is an empty class there are more parameters than classes and A is rank deficient. Choosing $P = S^{-1}$ gives a factorization $A = UP$, where each genotype class is represented in the matrix U by a row of all 0 except for a single 1 and the columns of U are orthogonal. The result is $U^T U = I \cdot D \cdot I = diag(n_1, n_2, \ldots, n_c)$ and $y^T U = [\Sigma_1 y \quad \Sigma_2 y \ldots \Sigma_c y]$. This gives the full model formula

$$\begin{aligned} RSS &= y^T y - \frac{(\Sigma_1 y)^2}{n_1} - \frac{(\Sigma_2 y)^2}{n_2} - \cdots - \frac{(\Sigma_c y)^2}{n_c} \\ &= \sum_{l=1}^{c} \left(\Sigma_l (y \odot y) - \frac{(\Sigma_l y)^2}{n_l} \right) \end{aligned} \tag{10}$$

which can be recognized as Equation 7. It should be noted that the same expression is obtained from *any* parametric model with the same number of non-redundant parameters as there are genotype classes.

Adding one continuous covariate to an otherwise full model: The formula for a full model plus one continuous, additive covariate q without

interactions is

$$\begin{aligned} RSS =& y^T y - \frac{(\Sigma_1 y)^2}{n_1} - \cdots - \frac{(\Sigma_c y)^2}{n_c} \\ & - \left(y^T q - \frac{\Sigma_1 y \cdot \Sigma_1 q}{n_1} - \ldots - \frac{\Sigma_c y \cdot \Sigma_c q}{n_c} \right)^2 \cdot \\ & \left(q^T q - \frac{(\Sigma_1 q)^2}{n_1} - \cdots - \frac{(\Sigma_c q)^2}{n_c} \right)^{-1} \\ =& \sum_{l=1}^{c} SS_{yy,l} - \frac{\left(\sum_{l=1}^{c} SS_{yq,l}\right)^2}{\sum_{l=1}^{c} SS_{qq,l}} \end{aligned} \tag{11}$$

where

$$SS_{yq,l} = \Sigma_l (y \odot q) - \frac{\Sigma_l y \cdot \Sigma_l q}{n_l} \tag{12}$$

which is the sum of cross-products of class l. In the case q is binary, $y^T q$ reduces to $\Sigma_{\langle 1 \rangle} y$, $q^T q$ to $n_{\langle 1 \rangle}$ and $\Sigma_l q$ to $n_{l,\langle 1 \rangle}$, the number of individuals in class l with covariate value 1.

2 QTL backcross without epistasis nor covariates. A special case of Equation 11 is a two-QTL backcross model without epistasis and without covariates. The one-QTL model is full and adding a second QTL can be seen as adding a binary covariate, except that the genotype at the second locus is included in the genotype class definition. It is possible to choose the matrix P such that $y^T U = \begin{bmatrix} \Sigma_{[1*]y} & \Sigma_{[2*]} y & \Sigma_{[*1]} y \end{bmatrix}$ and

$$\begin{aligned} U^T U = & \begin{bmatrix} n_{[1*]} & 0 & n_{[11]} \\ 0 & n_{[2*]} & n_{[21]} \\ n_{[11]} & n_{[21]} & n_{[*1]} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ \frac{n_{[11]}}{n_{[1*]}} & \frac{n_{[21]}}{n_{[2*]}} & 1 \end{bmatrix} \cdot \\ & \begin{bmatrix} n_{[1*]} & 0 & 0 \\ 0 & n_{[2*]} & 0 \\ 0 & 0 & n_{[*1]} - \frac{n_{[11]}^2}{n_{[1*]}} - \frac{n_{[21]}^2}{n_{[2*]}} \end{bmatrix} \begin{bmatrix} 1 & 0 & \frac{n_{[11]}}{n_{[1*]}} \\ 0 & 1 & \frac{n_{[21]}}{n_{[2*]}} \\ 0 & 0 & 1 \end{bmatrix} \end{aligned} \tag{13}$$

giving the final expression

$$\begin{aligned} RSS = & y^T y - \frac{\left(\Sigma_{[1*]} y\right)^2}{n_{[1*]}} - \frac{\left(\Sigma_{[2*]} y\right)^2}{n_{[2*]}} \\ & - \left(\Sigma_{[*1]} y - \frac{n_{[11]}}{n_{[1*]}} \Sigma_{[1*]} y - \frac{n_{[21]}}{n_{[2*]}} \Sigma_{[2*]} y \right)^2 \cdot \\ & \left(n_{[*1]} - \frac{n_{[11]}^2}{n_{[1*]}} - \frac{n_{[21]}^2}{n_{[2*]}} \right)^{-1} \end{aligned} \tag{14}$$

where the last term gives the decrease in RSS after adding a second QTL to the model, without including interactions.

Equation 14 can, if all $n_l > 0$, be expressed as

$$\begin{aligned} RSS &= y^T y \\ &- \frac{\left(\Sigma_{[11]}y\right)^2}{n_{[11]}} - \frac{\left(\Sigma_{[21]}y\right)^2}{n_{[21]}} - \frac{\left(\Sigma_{[12]}y\right)^2}{n_{[12]}} - \frac{\left(\Sigma_{[22]}y\right)^2}{n_{[22]}} \\ &+ \left(\frac{\Sigma_{[11]}y}{n_{[11]}} - \frac{\Sigma_{[21]}y}{n_{[21]}} - \frac{\Sigma_{[12]}y}{n_{[12]}} + \frac{\Sigma_{[22]}y}{n_{[22]}}\right)^2 \cdot \\ &\left(\frac{1}{n_{[11]}} + \frac{1}{n_{[21]}} + \frac{1}{n_{[12]}} + \frac{1}{n_{[22]}}\right)^{-1} \end{aligned} \tag{15}$$

where the first five terms constitute the exact formula for the full 2 QTL model and the last term gives the increase in RSS after removing the epistasis parameter. This form is obtained by using a $A = UP$ factorization such that $U^T U$ can be expressed as a diagonal matrix with a rank 1 modification, and inverting $U^T U$ using the Sherman-Morrison matrix inversion lemma. The form of Equation 15 is not used in the implementation of PERF, but is presented for illustration.

Adding two continuous covariates to a full model: Equation 11 can be extended to include a second covariate p without interactions. Using the sums of squares notation the formula is

$$\begin{aligned} RSS &= \sum_{l=1}^{c} SS_{yy,l} - \frac{\left(\sum_{l=1}^{c} SS_{yq,l}\right)^2}{\sum_{l=1}^{c} SS_{qq,l}} \\ &- \left(\sum_{l=1}^{c} SS_{yp,l} - \frac{\left(\sum_{l=1}^{c} SS_{yq,l}\right)\left(\sum_{l=1}^{c} SS_{pq,l}\right)}{\sum_{l=1}^{c} SS_{qq,l}}\right)^2 \cdot \\ &\left(\sum_{l=1}^{c} SS_{pp,l} - \frac{\left(\sum_{l=1}^{c} SS_{pq,l}\right)^2}{\sum_{l=1}^{c} SS_{qq,l}}\right)^{-1} \end{aligned} \tag{16}$$

where the first sum of c terms gives the RSS for the full model without covariates, the second to last term gives the reduction in RSS for adding the first covariate, and the last gives the reduction after adding the second covariate. The terms are directly available from the PERF matrix algebra implementation. The jth term is z_j^2/D_{jj} where, as defined earlier, z is the solution vector to the equation system $Lz = U^T W y$ and D is the diagonal matrix in the factorization $U^T W U = LDL^T$.

Adding one extra discrete covariate including interactions to any model: The RSS for any model after adding a binary covariate and interaction parameters between the covariate and all other parameters except the

mean can be computed by dividing the individuals into groups based on the covariate type and computing the RSS independently for each group, giving for a covariate with l possible values $RSS = RSS_{\langle 1 \rangle} + \ldots + RSS_{\langle l \rangle}$, where the subscripts denote the value of the covariate. In matrix algebra this corresponds to choosing U such that L, D and $U^T y$ are partitioned into orthogonal blocks, and factorizing the blocks separately.

Implementation

PERF is based on the LDL^T factorization algorithm and triangular system solver from [16]. and modify them to exploit the sparsity structure of $U^T W U$. With an appropriate choice of U, the leading p columns are orthogonal, and the corresponding off-diagonal elements of L are zero. The first p diagonal elements of D are copied directly from $U^T W U$, and the factorization starts with row $p+1$ of L. For $j \leq p$ and $k > p$, element (j,k) of L is equal to element (j,k) of $U^T W U$ divided by the jth diagonal element of D. The first p steps of the triangular system solver are reduced to simple divisions.

If A is rank deficient the model is inappropriate, and the computation should be interrupted. Monitoring the rank of $W^{\frac{1}{2}} A$ is easily done by checking the diagonal elements of the matrix D and performing a condition number estimation of L using e.g. the algorithm presented in [16]. A non-positive diagonal element of D implies rank deficiency. The condition estimator will detect near rank-deficiency in the (more rare) cases when the matrix is ill-conditioned despite that the diagonal elements of D are positive. An alternative method is to use pivoting during the matrix factorization, but that would destroy the simple structure of the problem and degrade performance.

The practical performance of PERF is sensitive to the implementation. Two implementation strategies have been tried, differing only in how memory was accessed when the individuals were sorted into genotype classes, and the second strategy was approximately 40% faster than the first. This behavior is not specific to the PERF algorithm, but rather a common effect seen due to the complex memory hierarchies found in modern computer architectures. In the fastest implementation the sorting is based on the genotype class codes. The d-digit codes can be seen as a set of d^g base g-numbers, and can be used as indices when storing class counts and phenotype sums in d^g-element vectors. For individual i the index is computed as $l = \sum_{j=1}^{j=d} \gamma_{ij} \cdot g^{i-1}$ where γ_{ij} is the x_j genotype minus 1, e.g. 0, 1 or 2 if $g = 3$. Then n_l is increased by 1 and y_i is added to phenotype sum l. To save time, $\gamma_{ij} g^{i-1}$ vectors are computed during the data preparation step and stored on disk. The extra space required is moderate since the terms are small and can be stored as 8-bit integers when $g < 5$. For models with a binary covariate including all covariate interactions,

an extra digit is added to the class code and the covariate type treated as the genotype at an extra locus for which $g = 2$.

If the number of QTL d is greatly increased, the number of genotype classes will be very large and the above strategy must be modified for an efficient implementation. The number of arithmetic operations required by PERF will still be very small compared with that of QR factorization. The results presented limit d to 4, which is appropriate given the size of the real data sets, considering the need to avoid overfitting.

PERF is comapred to the updated QR factorization technique presented in [24], implemented using the Lapack library routines `dgeqrf`, `dormqr` and `dgels` [2]. The relative efficiency of updating and `dgels` only, the library routine used in e.g. the QTL analysis software Pseudomarker [34], depends on m, k and the number of covariates. We only use updating when it is faster than `dgels`, which is the case for all data sets considered in all models with one covariate when analyzing intercross data, and 1 QTL models without QTL-covariate interaction for backcross data. This makes the comparison as favorable as possible for the traditional method in relation to PERF. To completely ensure safe handling of possibly rank deficient matrices, QR factorization with column pivoting, e.g. using the Lapack routine `dgeqp3`, should be performed. This slows down the computations. Neither `dgels` nor the updating algorithm performs column pivoting. We make this choice to mimic the approach of [34]. There a first, but not complete, rank check is performed by monitoring the diagonal elements of the matrix R from the QR factorization. The corresponding PERF approach is to only check the diagonal elements of D, not performing any rank estimation.

Results

The methods were tested on data from a 999 mice intercross population [4] and from a 256 mice backcross population [26]. Both datasets were downloaded from the QTL Archive at `http://www.jax.org` and are publicly available from there. In a preparatory step, a set of 32 complete information realizations of the genotype data were generated using the software Pseudomarker [30,34]. A test program, performing a fixed number of RSS evaluations for 34 different $1-4$ QTL models using PERF or updated QR was implemented in C, and the test program computing time was measured for all models.

Computing time dependence on the number of parameters k, backcross data: Figure 1 shows the computing time for evaluating RSS for 19 different 1-4 QTL models of the backcross data. Figure 2 shows the backcross dataset computing time for PERF in closer detail. The times are normalized

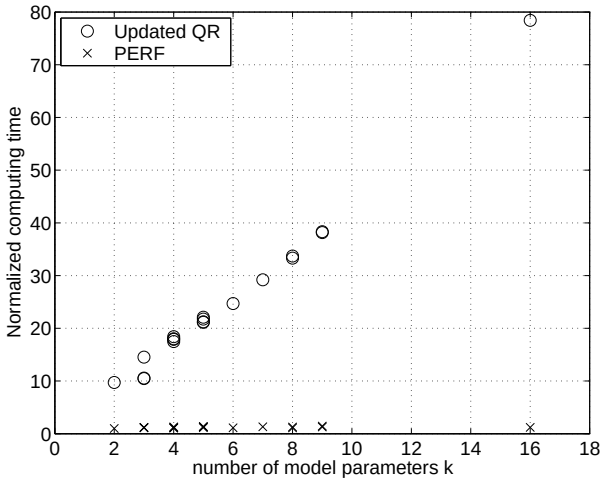

Fig. 1. The computing time for updated QR factorization and PERF, backcross data set. All results are normalized with the computing time for the $k = 2$ model using PERF.

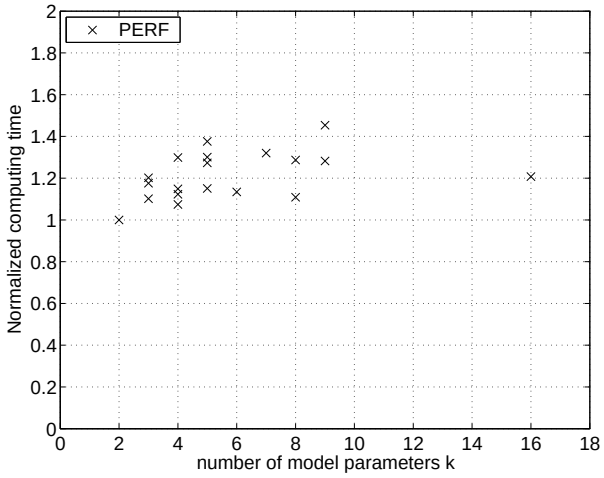

Fig. 2. Subsection of Figure 1. Computing time using PERF normalized with the time for the smallest model. The longest time is less than 1.5× that of the simplest model.

with the computing time for PERF applied on the two parameter (mean plus one QTL) model. The time for QR factorization increases greatly with the number of model parameters k, while the computing time for PERF is close to constant. The smallest difference in computing time between PERF and QR factorization occurs for the two parameter model, when QR factorization is 10 times as slow. The variation in QR factorization computing time for $k = 3$ demonstrates the gain of updating. The faster results are for models with 1 QTL and one additive covariate and the slower represents a model with 2 QTL without interactions nor covariates. There is sometimes a small variation in QR computing time for the same k also for other models, depending on the fact that it takes slightly longer to compute element-wise column products

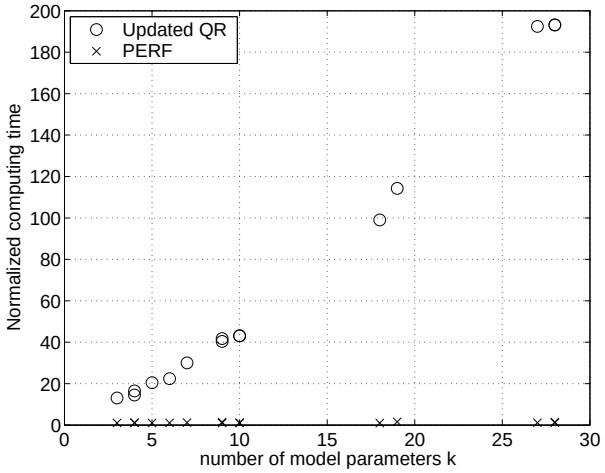

Fig. 3. The computing time for updated QR factorization and PERF, intercross data. All results are normalized with the computing time for the $k = 3$ model using PERF.

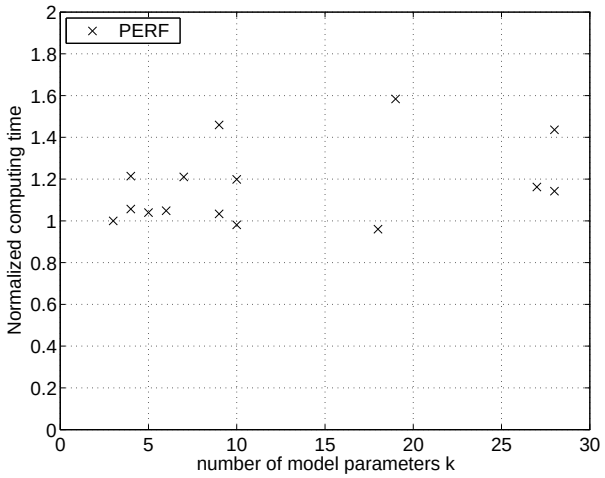

Fig. 4. Subsection of Figure 3. Computing time using PERF normalized with the time for the smallest model. The longest time is less than 1.6× that of the simplest model.

for interaction parameters when building the design matrix A, compared to only copying data from memory. The extra time required is however negligible compared to the total time. It can be observed in Figure 2 that the computing time for PERF also varies for the same number of parameters k. For example, when $k = 3$, a 1 QTL model with one additive, binary covariate is faster to compute than a 2 QTL model without covariate nor interactions. This is the result of differences in memory access time. The covariate information is stored in one n-element vector, while the genotypes are retrieved from a larger data structure.

Computing time dependence on the number of parameters k, intercross data: Figure 3 shows the intercross data computing time for computing

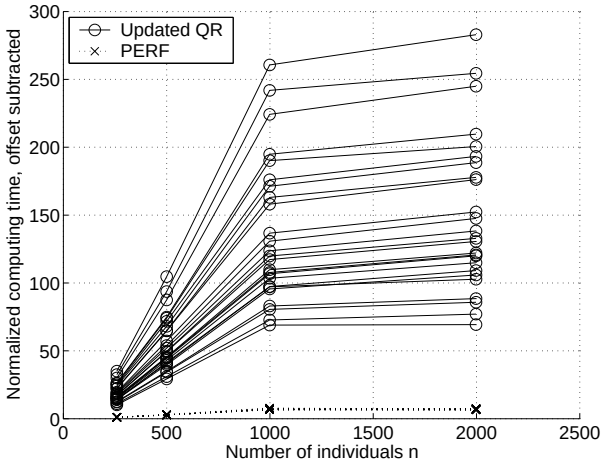

Fig. 5. Computing time after subtracting the offset, i.e. the time required for $n = 125$ using *the same* method, and dividing by the time for $n = 250$ using PERF. The computing time for both methods is super-linear in n.

RSS using 15 different 1-4 QTL models. Figure 4 shows the intercross dataset computing time for PERF of Figure 3 in closer detail. The times are normalized with the computing time for PERF applied on the three parameter (mean plus the additive and dominance effect of one QTL) model. The time for QR factorization increases greatly with the number of model parameters k, while again the computing time for PERF is close to constant. The smallest computing time difference between the two methods occurs for the $k = 3$ model, for which PERF is 13 times faster. The variation in computing time for PERF is small, and Figure 4 shows that the slowest computation takes less than $1.6\times$ the time of the fastest one. The "worst" case occurs for a three QTL model with pairwise, but not three-way, epistatic interactions. Then there are 10 non-orthogonal columns of U, and the time required to factorize U^TU is noticeable. Different computing times for the same k can be explained with differences in the amount of memory that needs to be accessed, or the sparsity pattern of U^TU.

The performance results presented are not considering the full process of minimizing the RSS as a function of $\bar{x}$. Figures 1 and 3 show the normalized computing time for a fixed number of RSS evaluations, not the total time to find the optimal $\bar{x}$ for the different models. However, in most settings, the time for evaluating the objective function is the main bottleneck in implementing an efficient minimization algorithm, with the execution time per function evaluation being essentially constant.

Time dependence on the number of individuals n: The computing time dependence on n was studied by comparing results for subsets and multiples of the original datasets. Figure 5 shows the computing time as a function of n. The time is normalized by first subtracting the offset, i.e. the respective computing times for a dataset with $n = 125$, and then dividing the results for

Model	Gain bc	(k)	Gain F2	(k)
1 QTL	10	(2)	13	(3)
2 QTL, -	12	(3)	20	(5)
3 QTL, -	14	(4)	25	(7)
4 QTL, -	16	(5)	29	(9)
2 QTL, 2w	16	(4)	39	(9)
3 QTL, 2w	24	(7)	72	(19)
3 QTL, 2w and 3w	29	(8)	168	(27)

Table 1
Relative computing time for updated QR compared to PERF for the same model, 999 individuals. Models without covariates. '-' indicates no interactions, '2w' pairwise epistatic interactions and '3w' three-way interactions.

both algorithms by the PERF computing time for $n = 250$. If the y-value in in Figure 5 is doubled when n is doubled this represents perfect linear dependence on n. Figure 5 shows that QR factorization is super-linear in n for small n, but the slope is smaller for larger n. If studying the plot in closer detail it can be seen that the same relation holds for PERF (not shown). The results show that the absolute increase in time is much greater for QR factorization when increasing n, however the relative increase is approximately the same.

Relative computing time QR/PERF: In some cases PERF enables the imputation method of [30] to compete with the regression method of [18] in terms of computational speed. The time required for a single QR factorization of a matrix of fixed size is the same regardless of the whether the entries represent complete genotype information, as when using the imputation method, or approximations, as with the regression method. When choosing the imputation method instead of regression, a single QR factorization using incomplete information is replaced by a number of RSS evaluations for complete genotype information. If using standard QR factorization for the imputed data sets, the total computing time is simply the time for regression multiplied with the number of imputations. Table 1 and Table 2 show how many RSS evaluations with PERF that can be performed during the time required for a single QR factorization in the case of 1-4 QTL models without covariates and 1-3 QTL models with one covariate. If the number of imputations chosen for a particular data set equals the number reported in Table 1 or Table 2, the imputation method is as fast as the regression method. If the number of imputations is smaller the imputation method is faster, and if the number of imputations is, for example, twice the number reported then imputation takes twice the time of the regression method. The number of imputations required varies with the type of data and the amount of missing information [30].

Model	Gain bc	(k)	Gain F2	(k)
1 QTL	10	(3)	12	(4)
2 QTL, -	16	(4)	n.a.	(6)
3 QTL, -	18	(5)	n.a.	(8)
2 QTL, 2w	16	(5)	36	(10)
3 QTL, 2w+3w	28	(9)	135	(28)

Table 2
Relative computing time for updated QR compared to PERF for the same model, 999 individuals. Models with one additive covariate. '-' indicates no interactions, '2w' pairwise epistatic interactions and '3w' three-way interactions.

The results in Tables 1 and 2 concern the time for evaluating the kernel function exclusively. Finding the most likely QTL positions $\bar{x}$ requires a search over all possible $\bar{x}$, e.g. using exhaustive stepwise search or a more advanced global optimization algorithm. Methods for global optimization were compared in [25], and there it was observed that the total computing time for the global search is close to directly proportional to the time of one RSS evaluation. This also holds when a more advanced search method than exhaustive grid search is used, and therefore the relative gain in computing time for the global search will be very close to the results shown in Tables 1 and 2.

It is common to observe several phenotypes in a single population, which gives a least squares problem with multiple right hand sides. Since the U^TWU factorization is independent of y, it is only necessary to compute new phenotype sums to evaluate the RSS for multiple phenotypes. The same technique can be applied to permuted data sets, when producing empirical significance thresholds [10]. Then also the covariate sums might need to be recomputed, depending on the permutation scheme that is employed. The standard method of QR factorization also allows for reusage, but the additional cost for each phenotype vector is greater when using the QR factorization.

The PERF algorithm is able to compute the RSS at least one order of magnitude faster than the most efficient QR factorization routine. The difference increases with an increasing number of individuals, and with more complicated models. This development opens the possibility for much more detailed data analysis in the same amount of time, and to perform more thorough model comparisons.

Acknowledgements

The authors wishes to thank Martina Persson, Lars Eldén, Dietrich von Rosen and Örjan Carlborg for helpful comments and discussion. The work of the first two authors has been funded by The Graduate School in Mathematics and Computing (FMB), Sweden.

References

[1] J. M. Álvarez-Castro and Ö. Carlborg, *A unified model for functional and statistical epistasis and its application in quantitative trait loci analysis*, Genetics **176** (2007), no. 2, 1151–1167.

[2] E. Anderson, Z. Bai, C. Bishop, J. Demmel, J. Dongarra, J. Du Croz, A. Greenbaum, S. Hammarling, A. McKenney, and D. Sorensen, *LAPACK: A portable linear algebra library for high-performance computers*, Tech. Report CS-90-105, Computer Science Department, University of Tennesse, Knoxville, 1990.

[3] R. Ball, *Bayesian methods for quantitative trait loci mapping based on model selection: Approximate analysis using the Bayesian information criterion*, Genetics **159** (2001), 1351–1364.

[4] W. Beamer, K. Shultz, G. Churchill, W. Frankel, D. Baylink, C. Rosen, and L. Donahue, *Quantitative trait loci for bone density in C57BL/6J and CAST/EiJ inbred mice*, Mammalian Genome **10** (1999), 1043–1049.

[5] Åke Björck, *Numerical methods for least squares problems*, SIAM, Society for Industrial and Applied Mathematics, Philadelphia, Pa., 1996.

[6] M. Bogdan, J. Ghosh, and R. Doerge, *Modifying the Schwarz Bayesian information criterion to locate multiple interacting quantitative trait loci*, Genetics **167** (2004), 989–999.

[7] K. Broman, *A review of statistical methods for QTL mapping in experimental crosses*, Lab Animal **30** (2001), 44–52.

[8] K. Broman and T. Speed, *A model selection approach for the identification of quantitative trait loci in experimental crosses*, Journal of the Royal Statistical Society: Series B **64** (2002), 641–656.

[9] Ö. Carlborg, L. Andersson, and B. Kinghorn, *The use of a genetic algorithm for simultaneous mapping of multiple interacting quantitative trait loci*, Genetics **155** (2000), 2003–2010.

[10] G. Churchill and R. Doerge, *Empirical threshold values for quantitative trait mapping*, Genetics **138** (1994), 963–971.

[11] C. Cockerham, *An extension of the concept of partitioning hereditary variance for analysis of covariance among relatives when epistasis is present*, Genetics **39** (1954), 859–882.

[12] B.E. Cooper, *A unifying computational method for the analysis of complete factorial experiments*, Communications of ACM **10** (1967), 27–34.

[13] R. Doerge, *Mapping and analysis of quantitative trait loci in experimental populations*, Nature Reviews Genetics **3** (2002), 43–52.

[14] W.T: Federer and M. Zelen, *Analysis of multifactor classifications with unequal numbers of observations*, Biometrics **22** (1966), 525–552.

[15] Ronald A. Fisher, *Statistical methods for research workers / by r.a. fisher*, 11th ed.(rev.) ed., Oliver & Boyd, Edinburgh :, 1950 (English).

[16] G. Golub and C. Van Loan, *Matrix Computations*, third ed., The Johns Hopkins University Press, 1996.

[17] I.J. Good, *The interaction algorithm and practica Fourier analysis*, Journal of the Royal Statistical Society, Series B **20** (1958), 361–372.

[18] C. Haley and S. Knott, *A simple regression method for mapping quantitative trait loci in line crosses using flanking markers*, Heredity **69** (1992), 315–324.

[19] C. Haley, S. Knott, and J.-M. Elsen, *Mapping quantitative trait loci in crosses between outbred lines using least squares*, Genetics **136** (1994), 1195–1207.

[20] R. Jansen and P. Stam, *High resolution of quantitative traits into multiple loci via interval mapping*, Genetics **136** (1994), 1447–1455.

[21] C.-H. Kao, Z.-B. Zeng, and R. Teasdale, *Multiple interval mapping for quantitative trait loci*, Genetics **152** (1999), 1203–1216.

[22] S. Knapp, W. Bridges, and D. Birkes, *Mapping quantitative trait loci using molecular marker linkage maps*, Theoretical and Applied Genetics **79** (1990), 583–592.

[23] E. Lander and D. Botstein, *Mapping mendelian factors underlying quantitative traits using RFLP linkage maps*, Genetics **121** (1989), 185–199.

[24] K. Ljungberg, S. Holmgren, and Ö. Carlborg, *Efficient algorithms for quantitative trait loci mapping problems*, Journal of Computational Biology **9** (2002), no. 6, 793–804.

[25] ———, *Simultaneous search for multiple QTL using the global optimization algorithm DIRECT*, Bioinformatics **20** (2004), 1887–1895.

[26] M. Mahler, C. Most, S. Schmidtke, J. Sundberg, R. Li, H. Hedrich, and G. Churchill, *Genetics of colitis susceptibility in IL-10-deficient mice: backcross versus F2 results contrasted by principal component analysis*, Genomics **80** (2002), 274–282.

[27] O. Martinez and R. Curnow, *Estimating the locations and the sizes of effects of quantitative trait loci using flanking markers*, Theoretical and Applied Genetics **85** (1992), 480–488.

[28] K. Mather and J. Jinks, *Biometrical Genetics*, Chapman and Hall, 1982.

[29] C. Nettelblad, S. Holmgren, L. Crooks, and Ö. Carlborg, *cnf2freq: Efficient determination of genotype and haplotype probabilities in outbred populations using markov models.*, BICoB 2008 (Sanguthevar Rajasekaran, ed.), Lecture Notes in Computer Science, vol. 5462, Springer, 2009, pp. 307–319.

[30] S. Sen and G. Churchill, *A statistical framework for quantitative trait mapping*, Genetics **159** (2001), 371–387.

[31] M. Sillanpää and J. Corander, *Model choice in gene mapping: what and why*, Trends Genet. **18** (2002), 301–307.

[32] M. Soller, T. Brody, and A. Genizi, *On the power of experimental design for the detection of linkage between marker loci and quantitative trait loci in crosses between inbred lines*, Theoretical and Applied Genetics **47** (1976), 35–39.

[33] A. Wright and R Mowers, *Multiple regression for molecular-marker, quantitative trait data from large F2 populations*, Theoretical and Applied Genetics **89** (1994), 305–312.

[34] H. Wu, S. Sen, K. Ljungberg, K. Broman, and G. Churchill, *Pseudomarker, version 2.01*, 2005, http://www.jax.org/staff/churchill/labsite/software/pseudomarker.

[35] F. Yates, *The design and analysis of factorial experiments*, Technical Communications of the Imperial Bureau of Soil Sciences **35** (1937).

[36] N. Yi, B. Yandell, G. Churchill, D. Allison, E. Eisen, and D. Pomp, *Bayesian model selection for genome-wide epistatic quantitative trait loci analysis*, Genetics **170** (2005), 1333–1344.

Appendix

This appendix lists expressions for RSS or $U^T y$ and $U^T U$ for common models. The matrix $U^T U$ is symmetric and only elements below and on the diagonal are given. As defined in the article, the following notation is used: At each locus, an individual has one of g genotypes, where $g = 2$ for a backcross and $g = 3$ for an intercross population. For any $\bar{x}$, the individuals will, depending on their genotypes at $\bar{x}$, belong to one of $c = g^d$ classes, where d is the number of QTL in the model. Each class is identified with a d-digit code without brackets (this is different from the article), where the jth digit denotes the genotype at x_j.The number of individuals in class l is denoted by n_l, and an asterix $*$ indicates a sum over all categories at that position. In the case of discrete covariate the division into covariate groups is done in the same way, but the covariate code is enclosed by angular brackets. The sum of elements in a vector v for individuals in a class is denoted by $\Sigma_l v$. The Hadamard (element-wise) product of two vectors is indicated by $\odot$. In formulas where the classes are identified by an index instead of a code, angular brackets are used around the index. Weights are not included in the formulas. In the case of weighted problems, the class counts n_l should be replaced by $\Sigma_l w$, $y^T y$ by $y^T W y$ and the phenotypes sums $\Sigma_l y$ by $\Sigma_l (Wy)$. The models are labelled according to the pattern *cross/number of QTL/interactions/covariate* where *cross* is either 'BC' for backcross, i.e. $g = 2$, or 'F2' for an intercross, $g = 3$, and *interactions* is either 'marg.' for a model with only marginal effects, '2way' for a model with marginal and all two-way genetic interaction effects, and '3way' for a model including all marginal, two-way and three-way genetic interaction effects. The last field *covariate* is either '-' indicating no covariates, 'cont.' for a continuous, additive covariate, 'bin.' for a binary, additive covariate or 'int.' for a binary covariate including interaction effects with all other parameters in the model. Binary covariates are included in the class definitions except for in Formula (20). Braces are used around class indices.

$$RSS = y^T y - \frac{(\Sigma_{\{1\}} y)^2}{n_{\{1\}}} - \frac{(\Sigma_{\{2\}} y)^2}{n_{\{2\}}} - \cdots - \frac{(\Sigma_{\{c\}} y)^2}{n_{\{c\}}} \tag{17}$$

$$RSS(\bar{x}) = y^T y - \frac{(\Sigma_{1*} y)^2}{n_{1*}} - \frac{(\Sigma_{2*} y)^2}{n_{2*}} - \left(\Sigma_{*1} y - \frac{n_{11}}{n_{1*}} \Sigma_{1*} y - \frac{n_{21}}{n_{2*}} \Sigma_{2*} y\right)^2 \cdot \left(n_{*1} - \frac{n_{11}^2}{n_{1*}} - \frac{n_{21}^2}{n_{2*}}\right)^{-1} \tag{18}$$

$$RSS(\bar{x}) = y^T y - \frac{\left(\Sigma_{\{1\}} y\right)^2}{n_{\{1\}}} - \cdots - \frac{\left(\Sigma_{\{c\}} y\right)^2}{n_{\{c\}}} - \left(y^T q - \frac{\Sigma_{\{1\}}(y \odot q)}{n_{\{1\}}} - \ldots - \frac{\Sigma_{\{c\}}(y \odot q)}{n_{\{c\}}}\right)^2 \cdot \left(q^T q - \frac{\Sigma_{\{1\}}(q \odot q)}{n_{\{1\}}} - \cdots - \frac{\Sigma_{\{c\}}(q \odot q)}{n_{\{c\}}}\right)^{-1} \tag{19}$$

$$RSS(\bar{x}) = y^T y - \frac{\left(\Sigma_{\{1\}} y\right)^2}{n_{\{1\}}} - \cdots - \frac{\left(\Sigma_{\{c\}} y\right)^2}{n_{\{c\}}} - \left(\Sigma_{\langle 1\rangle} y - \frac{\Sigma_{\{1\},\langle 1\rangle} y}{n_{\{1\}}} - \ldots - \frac{\Sigma_{\{c\},\langle 1\rangle} y}{n_{\{c\}}}\right)^2 \cdot \left(n_{\langle 1\rangle} - \frac{n_{\{1\},\langle 1\rangle}}{n_{\{1\}}} - \cdots - \frac{n_{\{c\},\langle 1\rangle}}{n_{\{c\}}}\right)^{-1} \tag{20}$$

Formula 18 with weighting is

$$\begin{aligned} RSS(\bar{x}) = y^T W y - \frac{\left(\Sigma_{[1*]}(Wy)\right)^2}{\Sigma_{[1*]} w} - \frac{\left(\Sigma_{[2*]}(Wy)\right)^2}{\Sigma_{[2*]} w} - \\ \left(\Sigma_{[*1]}(Wy) - \frac{\Sigma_{[11]} w}{\Sigma_{[1*]} w}\Sigma_{[1*]}(Wy) - \frac{\Sigma_{[21]} w}{\Sigma_{[2*]} w}\Sigma_{[2*]}(Wy)\right)^2 \cdot \\ \left(\Sigma_{[*1]} w - \frac{\left(\Sigma_{[11]} w\right)^2}{\Sigma_{[1*]} w} - \frac{\left(\Sigma_{[21]} w\right)^2}{\Sigma_{[2*]} w}\right)^{-1} \end{aligned} \tag{21}$$

which is also simple to evaluate. If all weights are equal to 1 formula 21 reduces to formula 18.

Formulas without covariates

BC/1/marg./- Formula (17)

BC/2/marg./- Formula (18)

BC/2/2way/- Formula (17)

BC/3/marg./-

$$\begin{pmatrix} n_{1**} & & & \\ 0 & n_{2**} & & \\ n_{11*} & n_{21*} & n_{*1*} & \\ n_{1*1} & n_{2*1} & n_{*11} & n_{**1} \end{pmatrix} \begin{pmatrix} \Sigma_{1**} y \\ \Sigma_{2**} y \\ \Sigma_{*1*} y \\ \Sigma_{**1} y \end{pmatrix} \tag{22}$$

BC/3/2way/-

$$\begin{pmatrix} n_{11*} & & & & & & \\ 0 & n_{21*} & & & & & \\ 0 & 0 & n_{12*} & & & & \\ 0 & 0 & 0 & n_{22*} & & & \\ 0 & 0 & n_{121} & n_{221} & n_{*21} & & \\ n_{111} & n_{211} & 0 & 0 & 0 & n_{*11} & \\ n_{111} & 0 & n_{121} & 0 & n_{121} & n_{111} & n_{1*1} \end{pmatrix} \begin{pmatrix} \Sigma_{11*}y \\ \Sigma_{21*}y \\ \Sigma_{12*}y \\ \Sigma_{22*}y \\ \Sigma_{*21}y \\ \Sigma_{*11}y \\ \Sigma_{1*1}y \end{pmatrix}$$

BC/3/3way/- Formula (17)

BC/4/marg./-

$$\begin{pmatrix} n_{1***} & & & & \\ 0 & n_{2***} & & & \\ n_{11**} & n_{21**} & n_{*1**} & & \\ n_{1*1*} & n_{2*1*} & n_{*11*} & n_{**1*} & \\ n_{1**1} & n_{2**1} & n_{*1*1} & n_{**11} & n_{***1} \end{pmatrix} \begin{pmatrix} \Sigma_{1***}y \\ \Sigma_{2***}y \\ \Sigma_{*1**}y \\ \Sigma_{**1*}y \\ \Sigma_{***1}y \end{pmatrix}$$

BC/5/marg./-

$$\begin{pmatrix} n_{1****} & & & & & \\ 0 & n_{2****} & & & & \\ n_{11***} & n_{21***} & n_{*1***} & & & \\ n_{1*1**} & n_{2*1**} & n_{*11**} & n_{**1**} & & \\ n_{1**1*} & n_{2**1*} & n_{*1*1*} & n_{**11*} & n_{***1*} & \\ n_{1***1} & n_{2***1} & n_{*1**1} & n_{**1*1} & n_{***11} & n_{****1} \end{pmatrix} \begin{pmatrix} \Sigma_{1****}y \\ \Sigma_{2****}y \\ \Sigma_{*1***}y \\ \Sigma_{**1**}y \\ \Sigma_{***1*}y \\ \Sigma_{****1}y \end{pmatrix}$$

F2/1/marg./- Formula (17)

F2/2/marg./-

$$\begin{pmatrix} n_{1*} & & & & \\ 0 & n_{2*} & & & \\ 0 & 0 & n_{3*} & & \\ n_{11} & n_{21} & n_{31} & n_{*1} & \\ n_{12} & n_{22} & n_{32} & 0 & n_{*2} \end{pmatrix} \begin{pmatrix} \Sigma_{1*}y \\ \Sigma_{2*}y \\ \Sigma_{3*}y \\ \Sigma_{*1}y \\ \Sigma_{*2}y \end{pmatrix}$$

F2/2/2way/- Formula (17)

F2/3/marg./-

$$\begin{pmatrix} n_{1**} & & & & & & \\ 0 & n_{2**} & & & & & \\ 0 & 0 & n_{3**} & & & & \\ n_{11*} & n_{21*} & n_{31*} & n_{*1*} & & & \\ n_{12*} & n_{22*} & n_{32*} & 0 & n_{*2*} & & \\ n_{1*1} & n_{2*1} & n_{3*1} & n_{*11} & n_{*21} & n_{**1} & \\ n_{1*2} & n_{2*2} & n_{3*2} & n_{*12} & n_{*22} & 0 & n_{**2} \end{pmatrix} \begin{pmatrix} \Sigma_{1**}y \\ \Sigma_{2**}y \\ \Sigma_{3**}y \\ \Sigma_{*1*}y \\ \Sigma_{*2*}y \\ \Sigma_{**1}y \\ \Sigma_{**2}y \end{pmatrix}$$

F2/3/2way/-

$$
\begin{pmatrix}
n_{11*} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & n_{112} & 0 \\
 & n_{21*} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & n_{212} & 0 \\
 & & n_{31*} & 0 & 0 & 0 & 0 & 0 & 0 & n_{312} & 0 \\
 & & & n_{12*} & 0 & 0 & 0 & 0 & 0 & 0 & n_{122} \\
 & & & & n_{22*} & 0 & 0 & 0 & 0 & 0 & n_{222} \\
 & & & & & n_{32*} & 0 & 0 & 0 & 0 & n_{322} \\
 & & & & & & n_{13*} & 0 & 0 & 0 & 0 \\
 & & & & & & & n_{23*} & 0 & 0 & 0 \\
 & & & & & & & & n_{33*} & 0 & 0 \\
 & & & & & & & & & n_{*12} & 0 \\
 & & & & & & & & & & n_{*22}
\end{pmatrix}
\begin{pmatrix}
0 & n_{111}-n_{113} & 0 & n_{111}-n_{113} & 0 & n_{111} & n_{112} & 0 \\
0 & 0 & n_{211}-n_{213} & n_{211}-n_{213} & 0 & n_{211} & 0 & n_{212} \\
0 & 0 & 0 & n_{311}-n_{313} & 0 & n_{311} & 0 & 0 \\
0 & n_{121}-n_{123} & 0 & 0 & n_{121}-n_{123} & n_{121} & n_{122} & 0 \\
0 & 0 & n_{221}-n_{223} & 0 & n_{221}-n_{223} & n_{221} & 0 & n_{222} \\
0 & 0 & 0 & 0 & n_{321}-n_{323} & n_{321} & 0 & 0 \\
n_{132} & n_{131}-n_{133} & 0 & 0 & 0 & n_{131} & n_{132} & 0 \\
n_{232} & 0 & n_{231}-n_{233} & 0 & 0 & n_{231} & 0 & n_{232} \\
n_{332} & 0 & 0 & 0 & 0 & n_{331} & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & n_{112} & n_{212} \\
0 & 0 & 0 & 0 & 0 & 0 & n_{122} & n_{222} \\
n_{*32} & 0 & 0 & 0 & 0 & 0 & n_{132} & n_{232} \\
 & n_{1*1}+n_{1*3} & 0 & n_{111}+n_{113} & n_{121}+n_{123} & n_{1*1} & 0 & 0 \\
 & & n_{2*1}+n_{2*3} & n_{211}+n_{213} & n_{221}+n_{223} & n_{2*1} & 0 & 0 \\
 & & & n_{*11}+n_{*13} & 0 & n_{*11} & 0 & 0 \\
 & & & & n_{*21}+n_{*23} & n_{*21} & 0 & 0 \\
 & & & & & n_{**1} & 0 & 0 \\
 & & & & & & n_{1*2} & 0 \\
 & & & & & & & n_{2*2}
\end{pmatrix}
\begin{pmatrix}
\Sigma_{11*}y \\ \Sigma_{21*}y \\ \Sigma_{31*}y \\ \Sigma_{12*}y \\ \Sigma_{22*}y \\ \Sigma_{32*}y \\ \Sigma_{13*}y \\ \Sigma_{23*}y \\ \Sigma_{33*}y \\ \Sigma_{*12}y \\ \Sigma_{*22}y \\ \Sigma_{*32}y \\ \Sigma_{1*1}y+\Sigma_{1*3}y \\ \Sigma_{2*1}y+\Sigma_{2*3}y \\ \Sigma_{*11}y+\Sigma_{*13}y \\ \Sigma_{*21}y+\Sigma_{*23}y \\ \Sigma_{**1}y \\ \Sigma_{1*2}y \\ \Sigma_{2*2}y
\end{pmatrix}
$$

F2/3/3way/- Formula (17)

F2/4/marg./-

$$\begin{pmatrix} n_{1***} & & & & & & & & & \\ 0 & n_{2***} & & & & & & & & \\ 0 & 0 & n_{3***} & & & & & & & \\ n_{11**} & n_{21**} & n_{31**} & n_{*1**} & & & & & & \\ n_{12**} & n_{22**} & n_{32**} & 0 & n_{*2**} & & & & & \\ n_{1*1*} & n_{2*1*} & n_{3*1*} & n_{*11*} & n_{*21*} & n_{**1*} & & & & \\ n_{1*2*} & n_{2*2*} & n_{3*2*} & n_{*12*} & n_{*22*} & 0 & n_{**2*} & & & \\ n_{1**1} & n_{2**1} & n_{3**1} & n_{*1*1} & n_{*2*1} & n_{**11} & n_{**21} & n_{***1} & & \\ n_{1**2} & n_{2**2} & n_{3**2} & n_{*1*2} & n_{*2*2} & n_{**12} & n_{**22} & 0 & n_{***2} \end{pmatrix} \begin{pmatrix} \Sigma_{1***}y \\ \Sigma_{2***}y \\ \Sigma_{3***}y \\ \Sigma_{*1**}y \\ \Sigma_{*2**}y \\ \Sigma_{**1*}y \\ \Sigma_{**2*}y \\ \Sigma_{***1}y \\ \Sigma_{***2}y \end{pmatrix}$$

Formulas including a covariate

BC/1/marg./int. Formula (17)

BC/1/marg./cont. Formula (19)

BC/1/marg./bin. Formula (18)

BC/2/marg./int.

$$\begin{pmatrix} n_{1*\langle m\rangle} & & & & & \\ 0 & n_{2*\langle m\rangle} & & & & \\ 0 & 0 & n_{1*\langle f\rangle} & & & \\ 0 & 0 & 0 & n_{2*\langle f\rangle} & & \\ 0 & 0 & n_{11\langle f\rangle} & n_{21\langle f\rangle} & n_{*1\langle f\rangle} & \\ n_{11\langle m\rangle} & n_{21\langle m\rangle} & 0 & 0 & 0 & n_{*1\langle m\rangle} \end{pmatrix} \begin{pmatrix} \Sigma_{1*\langle m\rangle}y \\ \Sigma_{2*\langle m\rangle}y \\ \Sigma_{1*\langle f\rangle}y \\ \Sigma_{2*\langle f\rangle}y \\ \Sigma_{*1\langle f\rangle}y \\ \Sigma_{*1\langle m\rangle}y \end{pmatrix}$$

BC/2/marg./cont.

$$\begin{pmatrix} n_{1*} & & & \\ 0 & n_{2*} & & \\ n_{11} & n_{21} & n_{*1} & \\ \Sigma_{1*}q & \Sigma_{2*}q & \Sigma_{*1}q & q^Tq \end{pmatrix} \begin{pmatrix} \Sigma_{1*}y \\ \Sigma_{2*}y \\ \Sigma_{*1}y \\ q^Ty \end{pmatrix}$$

BC/2/marg./bin. Formula (22)

BC/2/2way/int. Formula (17)

BC/2/2way/cont. Formula (19)

BC/2/2way/bin. Formula (20)

BC/3/marg./bin. Formula ()

BC/3/3way/int. Formula (17)

BC/3/3way/cont. Formula (19)

BC/3/3way/bin. Formula (20)

BC/4/marg./bin. Formula ()

F2/1/marg./int. Formula (17)

F2/1/marg./cont. Formula (19)

F2/1/marg./bin. Formula (20)

F2/2/marg./int.

$$\begin{pmatrix} n_{1*\langle m\rangle} & & & & & & & & & \\ 0 & n_{2*\langle m\rangle} & & & & & & & & \\ 0 & 0 & n_{3*\langle m\rangle} & & & & & & & \\ 0 & 0 & 0 & n_{1*\langle f\rangle} & & & & & & \\ 0 & 0 & 0 & 0 & n_{2*\langle f\rangle} & & & & & \\ 0 & 0 & 0 & 0 & 0 & n_{3*\langle f\rangle} & & & & \\ 0 & 0 & 0 & n_{11\langle f\rangle} & n_{21\langle f\rangle} & n_{31\langle f\rangle} & n_{*1\langle f\rangle} & & & \\ 0 & 0 & 0 & n_{12\langle f\rangle} & n_{22\langle f\rangle} & n_{32\langle f\rangle} & 0 & n_{*2\langle f\rangle} & & \\ n_{11\langle m\rangle} & n_{21\langle m\rangle} & n_{31\langle m\rangle} & 0 & 0 & 0 & 0 & 0 & n_{*1\langle m\rangle} & \\ n_{12\langle m\rangle} & n_{22\langle m\rangle} & n_{32\langle m\rangle} & 0 & 0 & 0 & 0 & 0 & 0 & n_{*2\langle m\rangle} \end{pmatrix} \begin{pmatrix} \Sigma_{1*\langle m\rangle}y \\ \Sigma_{2*\langle m\rangle}y \\ \Sigma_{3*\langle m\rangle}y \\ \Sigma_{1*\langle f\rangle}y \\ \Sigma_{2*\langle f\rangle}y \\ \Sigma_{3*\langle f\rangle}y \\ \Sigma_{*1\langle f\rangle}y \\ \Sigma_{*2\langle f\rangle}y \\ \Sigma_{*1\langle m\rangle}y \\ \Sigma_{*2\langle m\rangle}y \end{pmatrix}$$

F2/2/marg./cont.

$$\begin{pmatrix} n_{1*} & & & & & \\ 0 & n_{2*} & & & & \\ 0 & 0 & n_{3*} & & & \\ n_{11} & n_{21} & n_{31} & n_{*1} & & \\ n_{12} & n_{22} & n_{32} & 0 & n_{*2} & \\ \Sigma_{1*}q & \Sigma_{2*}q & \Sigma_{3*}q & \Sigma_{*1}q & \Sigma_{*2}q & q^Tq \end{pmatrix} \begin{pmatrix} \Sigma_{1*}y \\ \Sigma_{2*}y \\ \Sigma_{3*}y \\ \Sigma_{*1}y \\ \Sigma_{*2} \\ q^Ty \end{pmatrix}$$

F2/2/2way/int. Formula (17)

F2/2/2way/cont. Formula (19)

F2/2/2way/bin. Formula (20)

F2/3/3way/int. Formula (17)

F2/3/3way/cont. Formula (19)

F2/3/3way/bin. Formula (20)