Efficient evaluation of the residual sum of squares for quantitative trait locus models in the case of complete marker genotype information

Kajsa Ljungberg

Information Technology, Division of Scientific Computing, Uppsala University, 751 05 Uppsala, Sweden

ABSTRACT

Motivation: A core computation of many popular quantitative trait locus, QTL, mapping methods 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 the standard method QR factorization, but together the RSS computations take considerable time and often constitute the major part of the computational effort.

Results: We present an algorithm for RSS evaluation that is mathematically equivalent to evaluation via QR factorization but 10-100 times faster depending on the model and data dimensions. It can be used for all standard QTL models. Our method opens the possibility for more detailed data analysis and more extensive model comparisons.

Availability: C code, detailed derivations and general implementation strategies are available from the author on request.

Contact: kajsa.ljungberg@it.uu.se

Supplementary information: http://user.it.uu.se/˜kl/Appendix.pdf

INTRODUCTION

Established methods for mapping quantitative trait loci, QTL, include interval mapping (Lander and Botstein, 1989) and multiple interval mapping (Kao *et al.*, 1999). Reviews of statistical methods and computational challenges for QTL mapping in experimental crosses are given in (Doerge, 2002; Broman, 2001). Despite rapid development in hardware and algorithms, computational demand restricts the use of 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. (Carlborg *et al.*, 2000; Ball, 2001; Broman and Speed, 2002; Doerge, 2002; Sillanpää and Corander, 2002; Yi *et al.*, 2005). 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 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. (Soller *et al.*, 1976; Wright and Mowers, 1994; Broman and Speed, 2002; Bogdan *et al.*, 2004). Other authors propose multiple imputation for generating complete data realizations (Ball, 2001; Sen and Churchill, 2001), or replacing uncertain observations with a set of weighted complete observations (Jansen and Stam, 1994). For all these methods, computing the residual sum of squares, RSS, for a possibly weighted least squares problem constitutes a significant part of the computational effort.

In this paper we demonstrate how 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. 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 the expressions become more involved, but a general algorithm that can accommodate epistasis as well as discrete and continuous covariates can be defined. In this paper we present PERF, Pseudomarker Evaluation of RSS Function, a method for evaluating the RSS in the case of complete, possibly imputed, (pseudo-)marker genotype information. 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.

SYSTEM AND METHODS

Consider a dataset of n individuals. The phenotype observations are collected in a vector $y \in \mathrm{R}^{\mathrm{n}\times 1}$. A linear k-parameter model with d QTL is constructed,

$$y_i = \sum_{j=1}^{k} a_{ij} b_j + \epsilon_i \tag{1}$$

where a_{ij} is the indicator variable of individual i for the jth parameter, b_j is the regression coefficient, and ϵ_i is the error. Both additive and interaction variables may be included. The vector $\bar{x} \in \mathrm{R}^{\mathrm{d}\times 1}$ is a set of d putative QTL positions, where d ranges from 1 to at least 10. Assume that complete genotype information is available at $\bar{x}$. Given a model, genotype information and possibly covariate data, the design matrix $A(\bar{x}) \in \mathrm{R}^{\mathrm{n}\times \mathrm{k}}$ can be constructed. The observations may be weighted by W, a diagonal matrix of positive weights. The RSS is computed as

$$RSS(\bar{x}) = \min_b \left(A(\bar{x})b - y\right)^T W \left(A(\bar{x})b - y\right) \tag{2}$$

which is typically solved by QR factorization of $W^{1/2}A$. In this article we take the perspective of (Broman and Speed, 2002) where QTL mapping is viewed as a model selection problem and the focus therefore is on identifying QTL, while estimating precise effects is deferred until the QTL locations have been determined. For this purpose, only RSS and not the regression parameter vector b, is needed. We have used the model parameterization of (Mather and Jinks, 1982), but it is important to note that the RSS will be equal using any other model, e.g. from (Cockerham, 1954), where the parameters are linear combinations of the model of (Mather and Jinks, 1982). We will show how an alternative formulation of

Equation 2 leads to a highly efficient way of computing the RSS, substantially faster than QR factorization.

As indicated in Equation 2, the residual sum of squares is a function of the set of putative QTL positions $\bar{x}$. To determine the most probable QTL locations given a model, the RSS, or a weighted average of a set of RSS, should be minimized over all possible $\bar{x}$. This is a multi-dimensional global optimization problem that has been investigated in for example (Carlborg *et al.*, 2000; Ljungberg *et al.*, 2004) and will not be further discussed here. The dependency on $\bar{x}$ is therefore dropped from all expressions.

To prepare for the presentation of the new method, we introduce some notation and then review a simple example of how to compute RSS using the variance analysis framework. At each locus, an individual has one of g genotypes, where $g = 2$ for a backcross and $g = 3$ for an intercross population. As defined earlier, 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. 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 interactions, the number of parameters equals the number of classes, and therefore the class phenotype means can be estimated independently. The variance analysis RSS formula is then

$$RSS = \sum_{l=1}^{c} SS_{yy,l} \tag{3}$$

where

$$SS_{yy,l} = \Sigma_l (y - \mu_l) \odot (y - \mu_l) = \Sigma_l (y \odot y) - \frac{(\Sigma_l y)^2}{n_l} \tag{4}$$

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', but we describe how the algorithm to obtain general expressions can be easily implemented. Discrete and continuous covariates are easily included. Our algorithm PERF has the computational efficiency of Equation 3 and is completely general.

A necessary condition for the efficient algorithms to be applicable is that the individuals can be sorted into genotype classes. This is possible in a number of cases. 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 (Sen and Churchill, 2001; Ball, 2001). Then a set of complete (pseudo-)marker genotype information sets, consistent with the known data, is generated. The RSS is computed according to Equation 2, either for each data realization separately (Sen and Churchill, 2001) or for the combined data set using an augmented design matrix A_{aug} with complete genotype information and giving each of the n_{imp} imputed observations the weight $1/n_{imp}$ (Ball, 2001). The latter approach is, from an algorithmic viewpoint, related to a third method, namely the generalized linear finite mixture model method presented in (Jansen and Stam, 1994). This is another method to which PERF is applicable. In this case individuals with uncertain genotypes are represented by several rows in the design matrix, and the rows of the resulting A_{aug} are weighted according to the probability of each genotype. In contrast to the method of (Ball, 2001), 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 (Knapp *et al.*, 1990; Martinez and Curnow, 1992; Haley and Knott, 1992) and Bayesian QTL mapping methods (Satagopan *et al.*, 1996; Sillanpää and Arjas, 1998) 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. Efficient computational methods for interval mapping and the linear regression method are presented in (Ljungberg *et al.*, 2002).

ALGORITHM

To exploit the advantages of complete genotype information we use an alternative formulation to Equation 2,

$$\begin{aligned} RSS &= y^T W y - y^T W A (A^T W A)^{-1} A^T W y \\ &= y^T W y - y^T W U P (P^T U^T W U P)^{-1} P^T U^T W y \\ &= y^T W y - y^T W U (L D L^T)^{-1} U^T W y \\ &= y^T W y - y^T W U L^{-T} D^{-1} L^{-1} U^T W y \\ &= y^T W y - z D^{-1} z \end{aligned} \tag{5}$$

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 (Cockerham, 1954) or (Mather and Jinks, 1982) model the same matrix U can be obtained by using different matrices P. The computationally most expensive step when implementing Equation 5 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 we can 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. Our method to avoid 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 LDL^T factorization, and of the equation system $Lz = U^T W y$ when solving for z. Below we present matrix expressions used in the implementation of PERF, and in addition 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 a joker sign $*$ 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.

We now present explicit examples of Equation 5 for a few models. A comprehensive list of matrices for common QTL models is given 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 our method, since it is interesting to try many different models when analyzing a dataset, see e.g (Broman and Speed, 2002). 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. We only show examples of the simple case $W = I$, 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 $\Sigma_l(Wy)$.

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, we can 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. We get $U^T U = I \cdot D \cdot I = diag(n_1, n_2, \ldots, n_c)$ and $y^T U = [\Sigma_1 y \quad \Sigma_2 y \ldots \Sigma_c y]$. This gives the full model formula

$$\begin{aligned} RSS &= y^T y - \frac{(\Sigma_1 y)^2}{n_1} - \frac{(\Sigma_2 y)^2}{n_2} - \cdots - \frac{(\Sigma_c y)^2}{n_c} \\ &= \sum_{l=1}^{c} \left(\Sigma_l (y \odot y) - \frac{(\Sigma_l y)^2}{n_l} \right) \end{aligned} \tag{6}$$

which can be recognized as Equation 3. 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

$$\begin{aligned} RSS =& y^T y - \frac{(\Sigma_1 y)^2}{n_1} - \cdots - \frac{(\Sigma_c y)^2}{n_c} \\ & - \left(y^T q - \frac{\Sigma_1 y \cdot \Sigma_1 q}{n_1} - \cdots - \frac{\Sigma_c y \cdot \Sigma_c q}{n_c} \right)^2 \cdot \\ & \left(q^T q - \frac{(\Sigma_1 q)^2}{n_1} - \cdots - \frac{(\Sigma_c q)^2}{n_c} \right)^{-1} \\ =& \sum_{l=1}^{c} SS_{yy,l} - \frac{\left(\sum_{l=1}^{c} SS_{yq,l}\right)^2}{\sum_{l=1}^{c} SS_{qq,l}} \end{aligned} \tag{7}$$

where

$$SS_{yq,l} = \Sigma_l (y \odot q) - \frac{\Sigma_l y \cdot \Sigma_l q}{n_l} \tag{8}$$

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 7 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. We can choose the matrix P such that $y^T U = \left[\Sigma_{[1*]y} \quad \Sigma_{[2*]} y \quad \Sigma_{[*1]} y\right]$ and

$$\begin{aligned} U^T U = \begin{bmatrix} n_{[1*]} & 0 & n_{[11]} \\ 0 & n_{[2*]} & n_{[21]} \\ n_{[11]} & n_{[21]} & n_{[*1]} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ \frac{n_{[11]}}{n_{[1*]}} & \frac{n_{[21]}}{n_{[2*]}} & 1 \end{bmatrix} \cdot \\ \begin{bmatrix} n_{[1*]} & 0 & 0 \\ 0 & n_{[2*]} & 0 \\ 0 & 0 & n_{[*1]} - \frac{n_{[11]}^2}{n_{[1*]}} - \frac{n_{[21]}^2}{n_{[2*]}} \end{bmatrix} \begin{bmatrix} 1 & 0 & \frac{n_{[11]}}{n_{[1*]}} \\ 0 & 1 & \frac{n_{[21]}}{n_{[2*]}} \\ 0 & 0 & 1 \end{bmatrix} \end{aligned} \tag{9}$$

giving the final expression

$$\begin{aligned} RSS = y^T y &- \frac{\left(\Sigma_{[1*]} y\right)^2}{n_{[1*]}} - \frac{\left(\Sigma_{[2*]} y\right)^2}{n_{[2*]}} \\ - & \left(\Sigma_{[*1]} y - \frac{n_{[11]}}{n_{[1*]}} \Sigma_{[1*]} y - \frac{n_{[21]}}{n_{[2*]}} \Sigma_{[2*]} y \right)^2 \cdot \\ & \left(n_{[*1]} - \frac{n_{[11]}^2}{n_{[1*]}} - \frac{n_{[21]}^2}{n_{[2*]}} \right)^{-1} \end{aligned} \tag{10}$$

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

Equation 10 can, if all $n_l > 0$, be expressed as

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

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^TU can be expressed as a diagonal matrix with a rank 1 modification, and inverting U^TU using the Sherman-Morrison matrix inversion lemma. The form of Equation 11 is not used in the implementation of PERF, but is presented for illustration.

Adding two continuous covariates to a full model: Equation 7 can be extended to include a second covariate p without interactions. Using the sums of squares notation the formula is

$$\begin{aligned} RSS = \sum_{l=1}^{c} SS_{yy,l} - \frac{\left(\sum_{l=1}^{c} SS_{yq,l}\right)^2}{\sum_{l=1}^{c} SS_{qq,l}} \\ - \left(\sum_{l=1}^{c} SS_{yp,l} - \frac{\left(\sum_{l=1}^{c} SS_{yq,l}\right)\left(\sum_{l=1}^{c} SS_{pq,l}\right)}{\sum_{l=1}^{c} SS_{qq,l}}\right)^2 \cdot \\ \left(\sum_{l=1}^{c} SS_{pp,l} - \frac{\left(\sum_{l=1}^{c} SS_{pq,l}\right)^2}{\sum_{l=1}^{c} SS_{qq,l}}\right)^{-1} \end{aligned} \qquad (12)$$

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 $Lz = U^TWy$ and D is the diagonal matrix in the factorization $U^TWU = LDL^T$.

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^Ty are partitioned into orthogonal blocks, and factorizing the blocks separately.

IMPLEMENTATION

We base PERF on the LDL^T factorization algorithm and triangular system solver from (Golub and Van Loan, 1996). and modify them to exploit the sparsity structure of U^TWU. 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^TWU, 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^TWU 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 (Golub and Van Loan, 1996). 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.

We have observed that PERF performance is sensitive to the implementation. Two implementation strategies have been tried for this paper, 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. 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. In this study we have limited d to 4, which is appropriate given the size of the real data sets.

We compare PERF with the updated QR factorization technique presented in (Ljungberg *et al.*, 2002), implemented using the Lapack library routines dgeqrf, dormqr and dgels (Anderson *et al.*, 1990). The relative efficiency of updating and dgels only, the library routine used in e.g. the QTL analysis software Pseudomarker (Wu *et al.*, 2005), depends on m, k and the number of covariates. We only use updating when it is faster than dgels, which for the data sets used in this paper is 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 (Wu *et al.*, 2005). 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, and not perform any rank estimation.

RESULTS

The methods were tested on data from a 999 mice intercross population (Beamer *et al.*, 1999) and from a 256 mice backcross population (Mahler *et al.*, 2002). Both datasets were downloaded from The Jackson Laboratory 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 (Sen and Churchill, 2001; Wu *et al.*, 2005). A test program, performing a fixed number

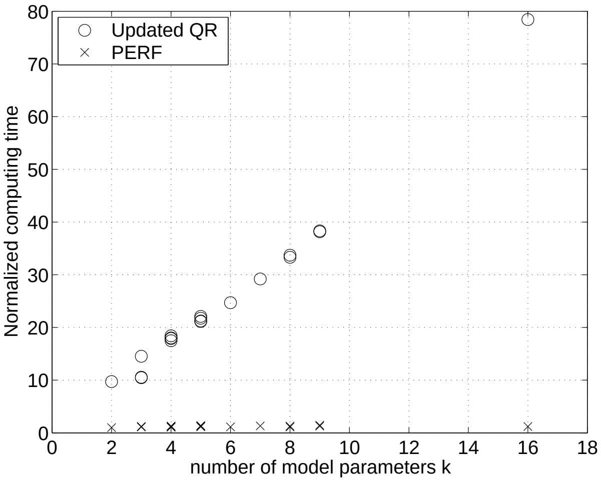

Fig. 1. The computing time for updated QR factorization and PERF, backcross data set. All results are normalized with the computing time for the $k = 2$ model using PERF.

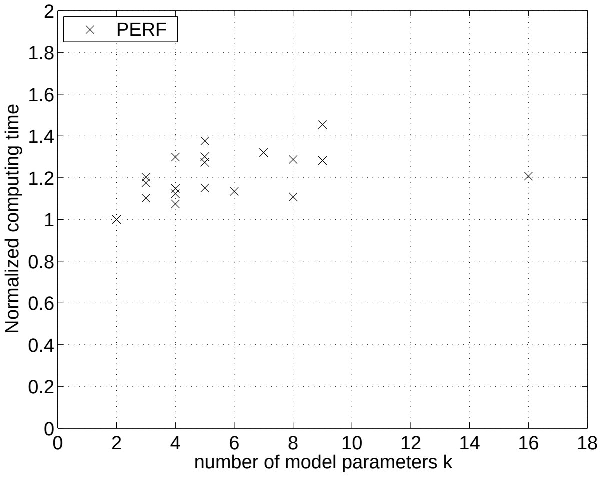

Fig. 2. Subsection of Figure 1. Computing time using PERF normalized with the time for the smallest model. The longest time is less than $1.5\times$ that of the simplest model.

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 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 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.

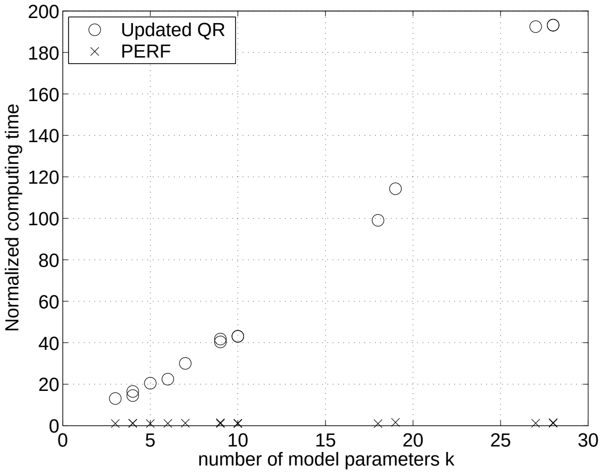

Fig. 3. The computing time for updated QR factorization and PERF, intercross data. All results are normalized with the computing time for the $k = 3$ model using PERF.

Computing time dependence on the number of parameters k, intercross data: Figure 3 shows the intercross data computing time for computing 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 $U^T U$ 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 $U^T U$.

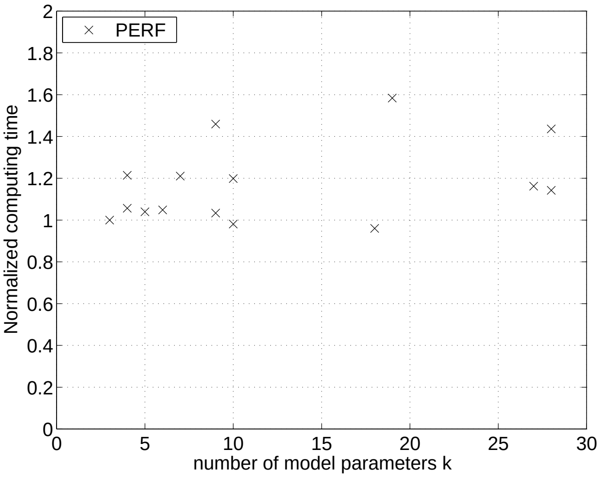

Fig. 4. Subsection of Figure 3. Computing time using PERF normalized with the time for the smallest model. The longest time is less than $1.6\times$ that of the simplest model.

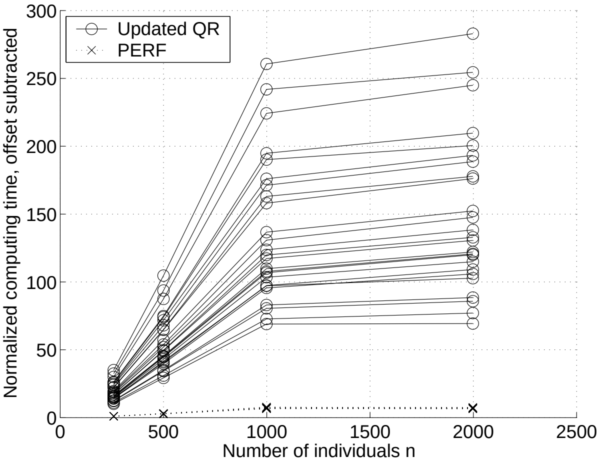

Fig. 5. Computing time after subtracting the offset, i.e. the time required for $n = 125$ using *the same* method, and dividing by the time for $n = 250$ using PERF. The computing time for both methods is super-linear in n.

We again point out that we in this paper do not minimize 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.

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 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.

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.

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)

Relative computing time QR/PERF: In some cases PERF enables the imputation method of (Sen and Churchill, 2001) to compete with the regression method of (Haley and Knott, 1992) 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 (Sen and Churchill, 2001).

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 (Ljungberg *et al.*, 2004), 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 holds also 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.

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.

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)

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 RSS for multiple phenotypes. The same technique can be applied to permuted data sets, when producing empirical significance thresholds (Churchill and Doerge, 1994). Then also the covariate sums must be recomputed. The standard method QR factorization also allows for reusage, but the additional cost for each phenotype vector is greater when using the QR factorization.

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

ACKNOWLEDGEMENTS

The author wishes to thank Martina Persson, Lars Eldén, Dietrich von Rosen, Örjan Carlborg and Sverker Holmgren for helpful comments and discussion. This work has been funded by The Graduate School in Mathematics and Computing (FMB), Sweden. The author is affiliated with The Linnaeus Center for Bioinformatics, Uppsala, Sweden.

REFERENCES

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

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

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

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

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

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

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

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

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

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

Golub,G. and Van Loan,C. (1996) *Matrix Computations*. The Johns Hopkins University Press, third edition.

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

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

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

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

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

Ljungberg,K., Holmgren,S. and Carlborg,Ö. (2002) Efficient algorithms for quantitative trait loci mapping problems. *Journal of Computational Biology*, **9**(6), 793–804.

Ljungberg,K., Holmgren,S. and Carlborg,Ö. (2004) Simultaneous search for multiple QTL using the global optimization algorithm DIRECT. *Bioinformatics*, **20**, 1887–1895.

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

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

Mather,K. and Jinks,J. (1982) *Biometrical Genetics*. Chapman and Hall, London, third edition.

Satagopan,J., Yandell,B., Newton,M. and Osborn,T. (1996) A Bayesian approach to detect quantitative trait loci using Markov chain Monte Carlo. *Genetics*, **144**, 805–816.

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

Sillanpää,M. and Arjas,E. (1998) Bayesian mapping of multiple quantitative trait loci from incomplete inbred line cross data. *Genetics*, **148**, 1373–1388.

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

Soller,M., Brody,T. and Genizi,A. (1976) 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**, 35–39.

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

Wu,H., Sen,S., Ljungberg,K., Broman,K. and Churchill,G. (2005) *Pseudomarker, Version 2.01*. http://www.jax.org/staff/churchill/labsite/software/ pseudomarker.

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

APPENDIX

Here we list the 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, we use the following notation. 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 a joker sign $*$ 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 (16). Braces are used around class indices.

$$RSS = y^T y - \frac{(\Sigma_{\{1\}} y)^2}{n_{\{1\}}} - \frac{(\Sigma_{\{2\}} y)^2}{n_{\{2\}}} - \cdots - \frac{(\Sigma_{\{c\}} y)^2}{n_{\{c\}}} \tag{13}$$

$$RSS(\bar{x}) = y^T y - \frac{(\Sigma_{1*} y)^2}{n_{1*}} - \frac{(\Sigma_{2*} y)^2}{n_{2*}} - \left(\Sigma_{*1} y - \frac{n_{11}}{n_{1*}} \Sigma_{1*} y - \frac{n_{21}}{n_{2*}} \Sigma_{2*} y\right)^2 \cdot \left(n_{*1} - \frac{n_{11}^2}{n_{1*}} - \frac{n_{21}^2}{n_{2*}}\right)^{-1} \tag{14}$$

$$\begin{aligned} RSS(\bar{x}) &= y^T y - \frac{\left(\Sigma_{\{1\}} y\right)^2}{n_{\{1\}}} - \cdots - \frac{\left(\Sigma_{\{c\}} y\right)^2}{n_{\{c\}}} \\ &- \left(y^T q - \frac{\Sigma_{\{1\}}(y \odot q)}{n_{\{1\}}} - \ldots - \frac{\Sigma_{\{c\}}(y \odot q)}{n_{\{c\}}}\right)^2 \cdot \left(q^T q - \frac{\Sigma_{\{1\}}(q \odot q)}{n_{\{1\}}} - \cdots - \frac{\Sigma_{\{c\}}(q \odot q)}{n_{\{c\}}}\right)^{-1} \end{aligned} \tag{15}$$

$$\begin{aligned} RSS(\bar{x}) &= y^T y - \frac{\left(\Sigma_{\{1\}} y\right)^2}{n_{\{1\}}} - \cdots - \frac{\left(\Sigma_{\{c\}} y\right)^2}{n_{\{c\}}} \\ &- \left(\Sigma_{\langle 1\rangle} y - \frac{\Sigma_{\{1\},\langle 1\rangle} y}{n_{\{1\}}} - \ldots - \frac{\Sigma_{\{c\},\langle 1\rangle} y}{n_{\{c\}}}\right)^2 \cdot \left(n_{\langle 1\rangle} - \frac{n_{\{1\},\langle 1\rangle}}{n_{\{1\}}} - \cdots - \frac{n_{\{c\},\langle 1\rangle}}{n_{\{c\}}}\right)^{-1} \end{aligned} \tag{16}$$

Formula 14 with weighting is Kolla!!!!!!!!

$$\begin{aligned} & RSS(\bar{x}) = y^T W y - \frac{\left(\Sigma_{[1*]}(Wy)\right)^2}{\Sigma_{[1*]} w} - \frac{\left(\Sigma_{[2*]}(Wy)\right)^2}{\Sigma_{[2*]} w} - \\ & \left(\Sigma_{[*1]}(Wy) - \frac{\Sigma_{[11]} w}{\Sigma_{[1*]} w} \Sigma_{[1*]}(Wy) - \frac{\Sigma_{[21]} w}{\Sigma_{[2*]} w} \Sigma_{[2*]}(Wy)\right)^2 \cdot \\ & \left(\Sigma_{[*1]} w - \frac{\left(\Sigma_{[11]} w\right)^2}{\Sigma_{[1*]} w} - \frac{\left(\Sigma_{[21]} w\right)^2}{\Sigma_{[2*]} w}\right)^{-1} \end{aligned} \tag{17}$$

which is also simple to evaluate. If all weights are equal to 1 formula 17 reduces to formula 14.

Formulas without covariates

BC/1/marg./- Formula (13)
BC/2/marg./- Formula (14)
BC/2/2way/- Formula (13)

BC/3/marg./-

$$\begin{pmatrix} n_{1**} & & & \\ 0 & n_{2**} & & \\ n_{11*} & n_{21*} & n_{*1*} & \\ n_{1*1} & n_{2*1} & n_{*11} & n_{**1} \end{pmatrix} \begin{pmatrix} \Sigma_{1**}y \\ \Sigma_{2**}y \\ \Sigma_{*1*}y \\ \Sigma_{**1}y \end{pmatrix} \tag{18}$$

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 (13)
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 (13)
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 (13)
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/-

$$\left(\begin{array}{ccccccccccccccccccc}
n_{11*} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & n_{112} & 0 & 0 & n_{111}-n_{113} & 0 & n_{111}-n_{113} & 0 & n_{111} & n_{112} & 0 \\
 & n_{21*} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & n_{212} & 0 & 0 & 0 & n_{211}-n_{213} & n_{211}-n_{213} & 0 & n_{211} & 0 & n_{212} \\
 & & n_{31*} & 0 & 0 & 0 & 0 & 0 & 0 & n_{312} & 0 & 0 & 0 & 0 & n_{311}-n_{313} & 0 & n_{311} & 0 & 0 \\
 & & & n_{12*} & 0 & 0 & 0 & 0 & 0 & 0 & n_{122} & 0 & n_{121}-n_{123} & 0 & 0 & n_{121}-n_{123} & n_{121} & n_{122} & 0 \\
 & & & & n_{22*} & 0 & 0 & 0 & 0 & 0 & n_{222} & 0 & 0 & n_{221}-n_{223} & 0 & n_{221}-n_{223} & n_{221} & 0 & n_{222} \\
 & & & & & n_{32*} & 0 & 0 & 0 & 0 & n_{322} & 0 & 0 & 0 & 0 & n_{321}-n_{323} & n_{321} & 0 & 0 \\
 & & & & & & n_{13*} & 0 & 0 & 0 & 0 & n_{132} & n_{131}-n_{133} & 0 & 0 & 0 & n_{131} & n_{132} & 0 \\
 & & & & & & & n_{23*} & 0 & 0 & 0 & n_{232} & 0 & n_{231}-n_{233} & 0 & 0 & n_{231} & 0 & n_{232} \\
 & & & & & & & & n_{33*} & 0 & 0 & n_{332} & 0 & 0 & 0 & 0 & n_{331} & 0 & 0 \\
 & & & & & & & & & n_{*12} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & n_{112} & n_{212} \\
 & & & & & & & & & & n_{*22} & 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{array}\right)
\left(\begin{array}{c}
\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{array}\right)$$

F2/3/3way/- Formula (13)
F2/4/marg./-

$$\left(\begin{array}{ccccccccc}
n_{1***} & & & & & & & & \\
0 & n_{2***} & & & & & & & \\
0 & 0 & n_{3***} & & & & & & \\
n_{11**} & n_{21**} & n_{31**} & n_{*1**} & & & & & \\
n_{12**} & n_{22**} & n_{32**} & 0 & n_{*2**} & & & & \\
n_{1*1*} & n_{2*1*} & n_{3*1*} & n_{*11*} & n_{*21*} & n_{**1*} & & & \\
n_{1*2*} & n_{2*2*} & n_{3*2*} & n_{*12*} & n_{*22*} & 0 & n_{**2*} & & \\
n_{1**1} & n_{2**1} & n_{3**1} & n_{*1*1} & n_{*2*1} & n_{**11} & n_{**21} & n_{***1} & \\
n_{1**2} & n_{2**2} & n_{3**2} & n_{*1*2} & n_{*2*2} & n_{**12} & n_{**22} & 0 & n_{***2}
\end{array}\right)
\left(\begin{array}{c}
\Sigma_{1***}y \\ \Sigma_{2***}y \\ \Sigma_{3***}y \\ \Sigma_{*1**}y \\ \Sigma_{*2**}y \\ \Sigma_{**1*}y \\ \Sigma_{**2*}y \\ \Sigma_{***1}y \\ \Sigma_{***2}y
\end{array}\right)$$

Formulas including a covariate
BC/1/marg./int. Formula (13)
BC/1/marg./cont. Formula (15)
BC/1/marg./bin. Formula (14)
BC/2/marg./int.

$$\left(\begin{array}{cccccc}
n_{1*\langle m\rangle} & & & & & \\
0 & n_{2*\langle m\rangle} & & & & \\
0 & 0 & n_{1*\langle f\rangle} & & & \\
0 & 0 & 0 & n_{2*\langle f\rangle} & & \\
0 & 0 & n_{11\langle f\rangle} & n_{21\langle f\rangle} & n_{*1\langle f\rangle} & \\
n_{11\langle m\rangle} & n_{21\langle m\rangle} & 0 & 0 & 0 & n_{*1\langle m\rangle}
\end{array}\right)
\left(\begin{array}{c}
\Sigma_{1*\langle m\rangle}y \\ \Sigma_{2*\langle m\rangle}y \\ \Sigma_{1*\langle f\rangle}y \\ \Sigma_{2*\langle f\rangle}y \\ \Sigma_{*1\langle f\rangle}y \\ \Sigma_{*1\langle m\rangle}y
\end{array}\right)$$

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 (18)
BC/2/2way/int. Formula (13)
BC/2/2way/cont. Formula (15)
BC/2/2way/bin. Formula (16)
BC/3/marg./bin. Formula ()
BC/3/3way/int. Formula (13)
BC/3/3way/cont. Formula (15)
BC/3/3way/bin. Formula (16)
BC/4/marg./bin. Formula ()
F2/1/marg./int. Formula (13)
F2/1/marg./cont. Formula (15)
F2/1/marg./bin. Formula (16)
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 (13)
F2/2/2way/cont. Formula (15)
F2/2/2way/bin. Formula (16)
F2/3/3way/int. Formula (13)
F2/3/3way/cont. Formula (15)
F2/3/3way/bin. Formula (16)