BIOINFORMATICS

Vol. 26 ISMB 2010, pages i149–i157
doi:10.1093/bioinformatics/btq211

Inferring combinatorial association logic networks in multimodal genome-wide screens

Jeroen de Ridder[1,2,3,*], Alice Gerrits[4], Jan Bot[1,3], Gerald de Haan[4], Marcel Reinders[1,3] and Lodewyk Wessels[2,3,*]

[1]Delft Bioinformatics Lab, Delft University of Technology, 2628 CD Delft, [2]Bioinformatics and Statistics, Department of Molecular Biology, Netherlands Cancer Institute, 1066 CX Amsterdam, [3]Netherlands Bioinformatics Center, 6525 GA Nijmegen and [4]Department of Cell Biology, Section Stem Cell Biology, University Medical Center Groningen, University of Groningen, 9700 AD Groningen, the Netherlands

ABSTRACT

Motivation: We propose an efficient method to infer combinatorial association logic networks from multiple genome-wide measurements from the same sample. We demonstrate our method on a genetical genomics dataset, in which we search for Boolean combinations of multiple genetic loci that associate with transcript levels.

Results: Our method provably finds the global solution and is very efficient with runtimes of up to four orders of magnitude faster than the exhaustive search. This enables permutation procedures for determining accurate false positive rates and allows selection of the most parsimonious model. When applied to transcript levels measured in myeloid cells from 24 genotyped recombinant inbred mouse strains, we discovered that nine gene clusters are putatively modulated by a logical combination of trait loci rather than a single locus. A literature survey supports and further elucidates one of these findings. Due to our approach, optimal solutions for multi-locus logic models and accurate estimates of the associated false discovery rates become feasible. Our algorithm, therefore, offers a valuable alternative to approaches employing complex, albeit suboptimal optimization strategies to identify complex models.

Availability: The MATLAB code of the prototype implementation is available on: http://bioinformatics.tudelft.nl/ or http://bioinformatics.nki.nl/

Contact: m.j.t.reinders@tudelft.nl; l.wessels@nki.nl

1 INTRODUCTION

To explain complex biological phenomena it is of vital importance to measure—in the same sample—all relevant (complementary) biological variables, and to measure these at a genome-wide scale. For this reason, many *multimodal* screens have been performed that have complemented transcriptional profiling with, among others, copy number variation measurements, transcription factor binding assays, methylation status profiling or genotype calls (Bystrykh, 2005; Pollack *et al.*, 2002; Shames *et al.*, 2006; Visel *et al.*, 2009).

A common aim in analyzing these multimodal datasets is to find associations between the biological variables measured to infer their regulatory role. Consider, for instance, a study in which expression profiles and genome-wide genotype data were obtained in hematopoietic cells from a panel of fully homozygous recombinant inbred mouse strains (Fig. 1A). This 'genetical genomics' approach enables the determination of expression quantitative trait loci (eQTLs) characterized by strong associations between the genotype and the observed expression levels (Jansen and Nap, 2001; Schadt *et al.*, 2003). In the absence of a strong direct association between the genotype and gene expression, real multi-locus interactions may still be present, due to epistatic interaction (Frankel and Schork, 1996; Michaelson *et al.*, 2009). Such interactions may not be detectable as (marginal) direct associations between the genotype and gene expression (Fig. 1B).

To alleviate this, approaches which evaluate the joint association of multiple loci and a phenotype of interest are required. Several approaches have been proposed to attack this problem. These approaches differ mostly regarding the way the associations are modeled and the strategy employed to solve the combinatorial optimization problem. Some approaches (Manichaikul *et al.*, 2009; Wongseree *et al.*, 2009) follow what could be loosely termed a two-stage approach, where all two-locus models are first evaluated, which, in stage two, are used in a greedy search to yield multi-locus models. Approaches employing more advanced strategies to traverse the space of possible models are represented by a genetic programming approach (Nunkesser *et al.*, 2007) and Markov Chain Monte Carlo (MCMC) approaches associated with Bayesian analyses (Mukherjee *et al.*, 2009; Zhang and Liu, 2007). Since two-stage approaches have been demonstrated to be suboptimal (Evans *et al.*, 2006) and advanced search strategies such as MCMC are very sensitive to their implementation and parameter settings, and are not guaranteed to be optimal, an approach that finds a provably global solution to a selected model within reasonable time is highly desirable. Of particular interest is the method of Ljungberg *et al.* (2004) which is used for the pair-scan analysis that is available on the GeneNetwork on http://genenetwork.org. Ljungberg *et al.* (2004) stress the importance of performing a global search rather than relying on greedy searches by (pre)selecting markers based on their marginal effects. To deal with the computational complexity associated with such an optimization problem, the authors present a method to find global optima of a linear regression problem for up to three predictors that is fast enough to be employed in permutation procedures.

In contrast to the class of additive models employed by Ljungberg *et al.* (2004) (and many other approaches), we follow others (Kooperberg and Ruczinski, 2004; Mukherjee *et al.*, 2009; Nunkesser *et al.*, 2007) and employ Boolean combinatorial logic to explicitly incorporate interactions in the eQTL inference. To this end, we infer combinatorial association logic (CAL) networks that combine the observed genotypes through AND ($\wedge$), OR ($\vee$) and XOR

*To whom correspondence should be addressed.

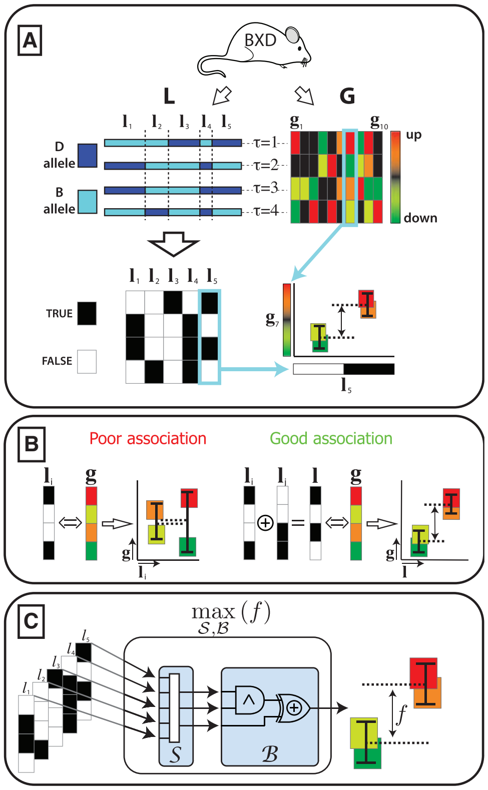


Fig. 1. Schematic overview of data and association inference. (**A**) A panel of BXD mice that is densely genotyped and expression profiled. The genotype data can be considered as binary vectors by choosing a binary encoding of the alleles (in the figure D=TRUE and B=FALSE) and putting thresholds that divide the genome into loci such that each locus differs in at least one element from its neighbors. The cartoon shows that good association is obtained between Locus 5 and Gene 7 because elevated expression is consistently observed in conjunction with the D allele of Locus 5. (**B**) Interaction among genetic features may destroy direct associations between individual loci and genes. The cartoon shows that configurations exist in which the gene expression can only be predicted by considering two loci simultaneously (using Boolean XOR logic). (**C**) By inferring CAL networks, interaction among genetic features is taken into account in the association inference. Inferring CAL networks is achieved by selecting the input loci with the selection function $\mathcal{S}$ and combining these with the appropriate Boolean function $\mathcal{B}$, such that the association (as measured by a scoring function f) between the network output and the gene of interest is maximized.

($\oplus$) functions by searching for associations between the result of the Boolean operation and the gene expression. The Boolean AND function can be used if altered expression is consistently observed in combination with a particular combination of two alleles (which do not necessarily have to be equal), but remains unchanged in all other genotype configurations. An example of a situation in which this may be observed is the case of two parallel pathways that only promote transcription of their downstream target when the genes in these pathways have specific alleles. Conversely, we may also consistently observe differential transcription in the strains for which either one of two loci is of a certain genotype. This may, for instance, be observed in case of a cascaded signaling pathway: a silencing mutation in one of the alleles can repress the entire pathway, regardless of which gene in the cascade contained this mutation. Boolean OR ($\vee$) and XOR ($\oplus$) are capable of capturing this behavior (Fig. 1B).

Like the search for optimal predictors in the additive model, inferring optimal predictors of a Boolean function is a challenging computational problem, especially considering that more complex combinations of these functions are also possible. Moreover, we noted that the objective function that needs to be optimized is highly discontinuous and nonlinear so that standard optimization techniques, such as genetic algorithms, simulated annealing and MCMC do not provide an optimal solution. Nevertheless, an efficient and—most importantly—global solution is highly desirable, since this allows permutation procedures with which significance estimates of the discovered associations can be realized (Ljungberg *et al.*, 2004).

In the following, we will mathematically prove that, under reasonable conditions, CAL network inference provides an efficient way to obtain globally optimal multi-locus models that associate multiple genomic loci with the expression of target genes. We illustrate our approach on the genetical genomics dataset from Gerrits *et al.* 2009, and using these data show that 100% accuracy is achieved at runtimes that are a fraction of those required for exhaustive search. Furthermore, we observe that using this approach complex associations are revealed that otherwise would have gone unnoticed. As such, our approach offers a useful alternative to the commonly used additive models and suboptimal search strategies.

2 METHODS

2.1 CAL network search

The construction of a CAL network that predicts the expression profile from a set of binary predictors can be formulated as an optimization problem. Interesting logic networks are those for which maximal association between the network output and the gene expression is obtained. Let $\mathbf{g}$ be the $(T\times 1)$ vector, with T the number of samples, containing the expression values of a gene and $\mathbf{L}$ the $(T\times L)$ matrix of binary predictors, e.g. the genotypes, where L is the number of predictors. A CAL network $\mathcal{L}$ is defined in terms of $\mathcal{S}(\mathbf{L};\mathbf{n}):\mathbb{B}^L\rightarrow\mathbb{B}^N$, a selection function that selects N columns from $\mathbf{L}$ and $\mathcal{B}(\mathbf{I}):\mathbb{B}^N\rightarrow\mathbb{B}$, a Boolean logic function that specifies the network topology. In the latter, $(T\times N)$ matrix $\mathbf{I}$ is a concatenation of the columns selected by $\mathcal{S}$, i.e. $\mathbf{I}=(\mathbf{i}_{\mathbf{n}(1)},\ldots,\mathbf{i}_{\mathbf{n}(N)})$, where $\mathbf{n}$ is a $(N\times 1)$ vector containing the indices of the selected columns. Consequently, CAL network $\mathcal{L}$ maps the genotype matrix $\mathbf{L}$ to a $(T\times 1)$ output vector $\mathbf{y}$ as follows:

$$\mathbf{y}=\mathcal{L}(\mathbf{L};\mathcal{B},\mathbf{n})=\mathcal{B}\big(\mathcal{S}(\mathbf{L};\mathbf{n})\big). \tag{1}$$

The association between $\mathbf{g}$ and $\mathbf{y}$ is quantified with an association measure $f(\mathbf{g},\mathbf{y})$

$$f(\mathbf{g},\mathbf{y})=\begin{cases}\dfrac{|\bar{\mathbf{x}}_0-\bar{\mathbf{x}}_1|}{\sqrt{\frac{(n_0-1)s_0^2+(n_1-1)s_1^2}{n_0+n_1-2}\left(\frac{1}{n_0}+\frac{1}{n_1}\right)}} & \text{if } (n_0>\eta)\cup(n_1>\eta)\\ 0 & \text{otherwise.}\end{cases} \tag{2}$$

For notational convenience, we used $\mathbf{x}_0=\{\mathbf{g}(\tau):\mathbf{y}(\tau)=0,\forall\tau\in(1,\ldots,T)\}$ and $\mathbf{x}_1=\{\mathbf{g}(\tau):\mathbf{y}(\tau)=1,\forall\tau\in(1,\ldots,T)\}$, i.e. vector $\mathbf{g}$ is split into $\mathbf{x}_0$ and $\mathbf{x}_1$ according to the Boolean values in $\mathbf{y}$. Furthermore, $\bar{\mathbf{x}}_0$ ($\bar{\mathbf{x}}_1$), s_0^2 (s_1^2) and n_0 (n_1) are defined as the sample mean, the sample variance and the number of elements in x_0 (x_1), respectively. Note that Equation (2) is equal to the absolute value of the t-statistic, except when n_0 or n_1 becomes too small, which ensures high f-values are only obtained in Case x_0 and x_1 have at least η elements.

The inference of CAL networks is a computationally challenging problem. Primarily, because the feature selection problem, i.e. finding the optimal vector $\mathbf{n}$, critically depends on the number of features that are considered.

In the case of genetic markers, this easily runs in the several hundreds to thousands. Moreover, the optimal subset of markers is heavily dependent on how these markers are combined, i.e. dependent on the optimal Boolean function $\mathcal{B}$. All together, one frequently has to rely on greedy search strategies that easily get stuck in local optima or near exhaustive searches that are computationally too expensive, especially when employed in permutation procedures required to assess statistical significance.

Our solution to this problem hinges upon two observations. First, in most practical datasets the sample size is relatively small, especially when compared to the number of features. This means that we can limit ourselves to considering only small CAL networks with few inputs, since larger networks are prone to overfitting, which makes them less informative. For this reason, and because most networks have many equivalent topologies that do not need to be evaluated due to symmetry, the set containing all unique and meaningful network topologies $\{\mathcal{B}_j : j=1,2,\cdots\}$ is relatively small (in the order of 10–100, depending on the desired topology). Consequently, the set of optimal input vectors $\{\mathbf{n}_j^* : j=1,2,\cdots\}$, associated with each $\mathcal{B}_j$, can be found by fixing $\mathcal{B}_j$ and maximizing for each $\mathcal{B}_j$ separately

$$\mathbf{n}_j^* = \underset{\mathbf{n}}{\operatorname{argmax}} \left\{ f\left(\mathbf{g}, \mathcal{B}_j(\mathcal{S}(\mathbf{L};\mathbf{n}))\right)\right\}. \qquad (3)$$

Second, we observe that Equation (3) still represents a complex optimization problem that can be significantly simplified by employing an approximation to the association measure, denoted by $\hat{f}$. In the following, we show that maximizing $\hat{f}$ is equivalent to maximizing f, but the maximization of the former can be very efficiently realized by using a branch and bound search. Before defining $\hat{f}$, we define the Boolean vector $\mathbf{y}^{\text{opt}}$ as the solution for which f reaches a global maximum independent of the network topology, i.e. $\mathbf{y}^{\text{opt}} = \operatorname{argmax}_{\mathbf{y}} f(\mathbf{g},\mathbf{y})$. Note that $\mathbf{y}^{\text{opt}}$ can be easily determined by sorting the gene expression vector $\mathbf{g}$ and evaluating all positions for a threshold t that splits $\mathbf{g}$ into x_0 and x_1 (Fig. 2A). For $\hat{f}$, we use the weighted Hamming similarity between $\mathbf{y}^{\text{opt}}$ and the network output $\mathbf{y}$

$$\hat{f}(\mathbf{y}^{\text{opt}},\mathbf{y}) = \sum_{\forall \tau} w(\tau) I(y^{\text{opt}}(\tau) = y(\tau)) \qquad (4)$$

where $w(\tau) > 0 \forall \tau$ denotes the weight for sample τ, and $I(\cdot)$ the indicator function, evaluating to ‘1’ if the τ-th element of vectors $\mathbf{y}^{\text{opt}}$ and $\mathbf{y}$ are equal.

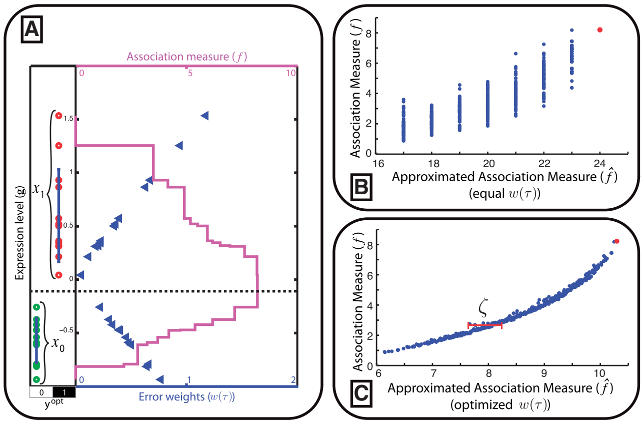


Fig. 2. Association versus approximated association. (**A**) Example gene expression vector (circles) split in x_0 and x_1 according to $\mathbf{y}_{\text{opt}}$. The magenta line denotes the association measure f, defined in Equation (2), as a function of a threshold t that splits the expression vector in x_0 and x_1. The blue triangles indicate the error weights $w(\tau)$ that result after optimizing them. (**B** and **C**) 500 random samples that are generated by introducing up to seven bit-flips in $\mathbf{y}^{\text{opt}}$ to show the relation between $\hat{f}$ and f. The red dot indicates $\hat{f}$ and f values for $\mathbf{y}^{\text{opt}}$. (B) shows the samples in case the weights are assumed equal. Although the trend of the data is monotonically increasing, a large spread around this trend is observed. (C) shows the same samples in case the weights are optimized, resulting in a near one-to-one relation between $\hat{f}$ and f.

For an example gene expression vector, Figure 2B shows 500 random samples of $(\hat{f}, f)$ pairs, in case all weights are equal to one. Although the trend of this distribution is monotonically increasing, the spread around the trend is substantial. This is undesirable because a maximum in $\hat{f}$ is only guaranteed to correspond to a maximum in f in case there is a direct one-to-one relation between them. Clearly, this is not the case in Figure 2B, since each value of $\hat{f}$ corresponds to many values of f. However, by optimizing the weights such that the difference between $\hat{f}$ and f is minimal, a near one-to-one relation can be obtained, as exemplified by Figure 2C. With the proper adjustments, detailed below, it is thus ensured that maximizing $\hat{f}$ is equivalent to maximizing f. The major advantage of maximizing $\hat{f}$ instead of f is that in the former each sample has an independent contribution to the association measure. This can be readily exploited using a branch and bound search, so that it is possible to avoid the expensive evaluation of the association measure.

2.2 Optimizing Equation (3)

Here, we show that optimizing Equation (3) can be achieved by first determining $\hat{f}^* = \max_{\mathbf{n}}(\hat{f})$, where $\hat{f}$ was defined in Equation (4). After this the search for $f^* = \max_{\mathbf{n}} f$ is readily solved by searching in the neighborhood of $\hat{f}^*$.

For a single sample τ, let $V^{(\tau)}$ be the set of input combinations such that $y(\tau) = y^{\text{opt}}(\tau) \forall \mathbf{n} \in V^{(\tau)}$, where $\mathbf{y} = \mathcal{L}(\mathbf{L};\mathcal{B},\mathbf{n})$.[1] Figure 3A–C shows how $V^{(\tau)}$ can be inferred from $\mathbf{L}$ and the truth table of $\mathcal{B}$. For a set of samples C, the input combinations $\mathbf{n} \in V^{(C)}$ for which all $\tau \in C$ reach the optimal output $\mathbf{y}^{\text{opt}}$ are found by taking the intersection of all the individual sets of input combinations, i.e. $V^{(C)} = \bigcap_{\tau \in C} V^{(\tau)}$. Note that, under the assumption that each sample has at least one non-zero locus, $V^{(\tau)} \neq \varnothing \forall \tau$. In other words, for individual samples there always exists a combination of inputs for which the network can reach the desired optimal output $\mathbf{y}^{\text{opt}}$. However, for an arbitrary combination of samples this is clearly not the case. If we observe that $V^{(C)} = \varnothing$, this means that for the collection of samples in C there does not exist a valid combination of inputs. Moreover, if $V^{(C)} = \varnothing$, all supersets of C will also result in the empty set. Finally we note that, by choosing a convenient binary encoding, $V^{(\tau)}$ and $V^{(C)}$ can be computed very efficiently by means of bitwise XNOR and AND operations, respectively (see Fig. 3D and the Supplementary Fig. S1 for details).

With these definitions in mind, we propose the following lemma:

LEMMA 1.

$$\hat{f}^* = \max_{C} \sum_{\forall \tau \in C} w(\tau) \qquad \text{subject to: } V^{(C)} \neq \varnothing \qquad (5)$$

PROOF. Let $C^* = \operatorname{argmax}_C \sum_{\forall \tau \in C} w(\tau)$, i.e. C^* is the set of solutions for which $\hat{f}^*$ is obtained. Since it is required that $V^{(C^*)} \neq \varnothing$, there must be at least one solution $\mathbf{n}$ such that $y^{\text{opt}}(\tau) = y(\tau) \forall \tau \in C^*$. Since for C^* the optimum in $\hat{f}$ is obtained, it must also hold that $y^{\text{opt}}(\tau) \neq y(\tau) \forall \tau \notin C^*$. This means that Equation (4) can be rewritten as follows: $\sum_{\forall \tau} w(\tau) I(y^{\text{opt}}(\tau) = y(\tau)) = \sum_{\forall \tau \in C} w(\tau)$, proving the statement in this lemma. ■

As argued by Lemma 1, Equation 4 is thus maximized by having as many samples in C as possible, while taking into account their respective weights $w(\tau)$.

Before we will show that Equation (5) fits a branch and bound framework, we first make the observation that for the relation between $\hat{f}$ and f the following holds:

$$(\hat{f}(\mathbf{y}^{\text{opt}},\mathbf{y}_1) < \hat{f}(\mathbf{y}^{\text{opt}},\mathbf{y}_2) - \zeta) \rightarrow (f(\mathbf{g},\mathbf{y}_1) < f(\mathbf{g},\mathbf{y}_2)), \qquad (6)$$

[1] Since we optimize Equation 3 for each $\mathcal{B}_j$ separately, we omit its subscript if its meaning is inconsequential.

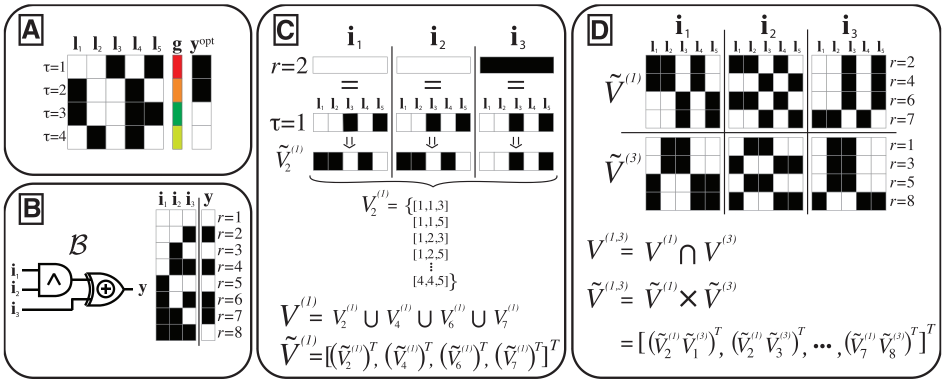

A
$\mathbf{l}_1$ $\mathbf{l}_2$ $\mathbf{l}_3$ $\mathbf{l}_4$ $\mathbf{l}_5$ $\mathbf{g}$ $\mathbf{y}^{opt}$
τ=1
τ=2
τ=3
τ=4
B
$\mathcal{B}$
$\mathbf{i}_1$ $\mathbf{i}_2$ $\mathbf{i}_3$
$\mathbf{y}$
r=1
r=2
r=3
r=4
r=5
r=6
r=7
r=8
C
$\mathbf{i}_1$ $\mathbf{i}_2$ $\mathbf{i}_3$
r=2
=
τ=1
⇓
$\tilde{V}_2^{(1)}$
$V_2^{(1)} = \{[1,1,3]$ $[1,1,5]$ $[1,2,3]$ $[1,2,5]$ ⋮ $[4,4,5]\}$
$V^{(1)} = V_2^{(1)} \cup V_4^{(1)} \cup V_6^{(1)} \cup V_7^{(1)}$
$\tilde{V}^{(1)} = [(\tilde{V}_2^{(1)})^T, (\tilde{V}_4^{(1)})^T, (\tilde{V}_6^{(1)})^T, (\tilde{V}_7^{(1)})^T]^T$
D
$\tilde{V}^{(1)}$
r=2
r=4
r=6
r=7
$\tilde{V}^{(3)}$
r=1
r=3
r=5
r=8
$V^{(1,3)} = V^{(1)} \cap V^{(3)}$
$\tilde{V}^{(1,3)} = \tilde{V}^{(1)} \times \tilde{V}^{(3)}$
$= [(\tilde{V}_2^{(1)} \tilde{V}_1^{(3)})^T, (\tilde{V}_2^{(1)} \tilde{V}_3^{(3)})^T, \cdots, (\tilde{V}_7^{(1)} \tilde{V}_8^{(3)})^T]^T$

Fig. 3. Computation of solution sets for each sample. **(A)** Example data from Figure 1A. **(B)** The topology and the truth table of the Boolean function $\mathcal{B}$ under investigation. **(C)** Explanation by example of the calculation of $V^{(\tau)}$, the set of all possible input combinations to $\mathcal{B}$ such that $y^{\text{opt}}(\tau) = y(\tau)$. This panel shows how $V^{(1)}$ is determined. Since $y^{\text{opt}}(1) = 1$, the rows from the truth table for which $y = 1$ are applicable, i.e, $r = \{2,4,6,7\}$. According to $r = 2$, the desired output for $\tau = 1$ is obtained by selecting any of the loci that are '0' for inputs $\mathbf{i}_1$ and $\mathbf{i}_2$, and loci that are '1' for input $\mathbf{i}_3$. Accordingly, for $\mathbf{i}_1$ we may select from the set: $\{\mathbf{l}_1, \mathbf{l}_2, \mathbf{l}_4\}$. This can be efficiently calculated by taking the XNOR (evaluates to '1' when both inputs are equal) between row $\tau = 1$ from the data matrix and the row $r = 2$ from the truth table, as shown in (**C**). Observe that the result is an efficient encoding of all the possible input combinations that satisfy $y^{\text{opt}}(1)$ while using $r = 2$ from the truth table. In general, we denote this set by $V_r^{(\tau)}$, and its binary encoding by $\tilde{V}_r^{(\tau)}$. To determine the complete set of valid input combinations for $\tau = 1$, rows 4, 6 and 7 need to be considered in a similar fashion. $V^{(1)}$ is now determined by taking the union of the subsets, i.e. $V^{(1)} = V_2^{(1)} \cup V_4^{(1)} \cup V_6^{(1)} \cup V_7^{(1)}$, which, in binary form, may be represented by a concatenation of $\tilde{V}_2^{(1)}$, $\tilde{V}_4^{(1)}$, $\tilde{V}_6^{(1)}$ and $\tilde{V}_7^{(1)}$. **(D)** This panel shows the valid input combinations for $\tau = 1$ and $\tau = 3$ in binary representation (i.e. $\tilde{V}^{(1)}$ and $\tilde{V}^{(3)}$). For any set of samples C the input combinations for which the output equals $\mathbf{y}^{\text{opt}}$ can be obtained by taking the intersection of the individual sets. In binary representation, this is equivalent to taking the row-wise cartesian product (row-wise product of all combinations of rows), as is shown in the panel.

where $\mathbf{y}_1$ and $\mathbf{y}_2$ are two Boolean vectors. Note that, for $\zeta = 0$, Equation (6) reduces to the requirement for strict monotonicity, and that for larger $\zeta > 0$ this requirement is increasingly relaxed. Even though this seems trivial, the value of this relation becomes clear by considering that if there exists a strong positive correlation between $\hat{f}$ and f, there may in fact exist a small ζ for which Equation (6) is true.

Based on Lemma 1 and Equation (6), we observe that solutions that are suboptimal in terms of $\hat{f}$ may still be optimal in terms of f, since ζ can be non-zero. In the following, let $\{\mathbf{y}_i : i = 1,2,\cdots\}$ and $\{C_i : i = 1,2,\cdots\}$ be all the network outputs and the sample sets for the solutions for which holds that $\hat{f}^* - \zeta \leq \hat{f}(\mathbf{y}^{\text{opt}}, \mathbf{y}_i) \leq \hat{f}^*$, respectively. Finally, let ζ be chosen such that Equation (6) holds. Our main theorem can now be formulated as follows:

THEOREM 2.

$$\mathbf{n}^* \in \bigcup_{\forall C_i} V^{(C_i)} \tag{7}$$

PROOF. First, assume that Equation (6) holds for $\zeta = 0$, and thus $\hat{f}(\mathbf{y}^{\text{opt}}, \mathbf{y}_i) = \hat{f}^* \, \forall i$. Furthermore, from Equation (6) it follows that in this case there exists a direct one-to-one relation between $\hat{f}$ and f. Consequently, a maximum in $\hat{f}$ is guaranteed to correspond to a maximum in f and $V^{(C_i)}$ must contain $\mathbf{n}^*$. This is true because from Lemma 1 it follows that $V^{(C_i)} \neq \varnothing$. For non-zero values of ζ, the one-to-one relation does not hold. However, from Equation (6), it follows that all values of f for which the corresponding $\hat{f}$ lies outside the interval $[\hat{f}^* - \zeta, \hat{f}^*]$ are strictly smaller than the value of f corresponding to $\hat{f}^*$. Thus, it must be the case that the maximum of f is constrained to solutions for which $\hat{f}$ lies in the interval $[\hat{f}^* - \zeta, \hat{f}^*]$. Therefore, the union of the sets of solutions that lie in this interval will contain $\mathbf{n}^*$. ■

From Theorem 2 it naturally follows that:

COROLLARY 3.

$$\mathbf{n}^* = \underset{\mathbf{n}}{\operatorname{argmax}} f(\mathbf{g}, \mathcal{L}(\mathbf{L}, \mathcal{B}, \mathbf{n})) \, \forall \mathbf{n} \in V^{(Q)}, \tag{8}$$

where $V^{(Q)} = \bigcup_{\forall C_i} V^{(C_i)}$. Notably, if there exists a small ζ for which Equation (6) holds, the number of solutions in $V^{(Q)}$ is limited, and hence $\mathbf{n}^*$ is easily determined by an exhaustive search over all possible solutions in $V^{(Q)}$. In the following, we show that in practice the set $V^{(Q)}$ is small by choosing $\mathbf{w}$ such that ζ is small.

2.2.1 *Estimating the weights*

Ideally, vector $\mathbf{w}$ is chosen such that ζ is minimal. For practical purposes, it is sufficient to choose $\mathbf{w}$ so that ζ is small, which can be realized by minimizing the difference between $\hat{f}$ and f. For this purpose, we sample the $(\hat{f}, f)$ relation by generating N random instances $\mathbf{y}_n$ by introducing up to m random bit-flips in $\mathbf{y}^{\text{opt}}$ (shown in Fig. 2B and C). The N corresponding association measures f_n and Hamming similarities are collected in vector $\mathbf{f} = [f(\mathbf{g}, \mathbf{y}_1), f(\mathbf{g}, \mathbf{y}_2), \cdots]^T$ and matrix $\hat{\mathbf{F}} = [(\mathbf{y}^{\text{opt}} \leftrightarrow \mathbf{y}_1)^T, (\mathbf{y}^{\text{opt}} \leftrightarrow \mathbf{y}_2)^T, \cdots]^T$, respectively. In the latter, $\leftrightarrow$ denotes the XNOR operation, which evaluates to '1' in case its arguments are equal. Notably, m (the number of bit-flips) should be chosen such that the region of interest of the distribution of f is sampled. Since we are interested only in network outputs that associate well with the gene expression, we can choose m rather small to focus only on the right tail for which a good fit between $\hat{f}$ and f is obtained. We found that smaller residuals were obtained by converting log-transformed f-values to z-scores, i.e. $\tilde{\mathbf{f}} = z(\ln \mathbf{f})$. Furthermore, to deal with the intercept, the matrix $\mathbf{F}$ is mean centered, denoted by $\tilde{\mathbf{F}}$. Using the vector $\tilde{\mathbf{f}}$ and matrix $\tilde{\mathbf{F}}$ we can find the weights $\mathbf{w}$ by constraint linear least squares minimization

$$\mathbf{w} = \underset{\mathbf{w}}{\operatorname{argmin}} ||\tilde{\mathbf{f}} - \tilde{\mathbf{F}}\mathbf{w}||_2, \qquad \text{subject to: } w(\tau) \geq w_\varepsilon \tag{9}$$

where $w_\varepsilon > 0$ is a small scalar that ensures each sample receives a non-zero weight. Figure 2 illustrates a typical example showing that the trend of the relation is monotonically increasing, and the spread around the trend is marginal, indicating that Equation (6) indeed holds for a small ζ.

2.2.2 *Estimating ζ*

The parameter ζ can be estimated by randomly resampling the $(\hat{f}, f)$ relation using the obtained weights and measuring the spread around the trend in the data in the $\hat{f}$ direction (Fig. 2C illustrates this schematically). To this end, lowess smoothing was performed to obtain the the trend in the data (Cleveland, 1979). Subsequently, the spread around this trend was obtained by applying a sliding window in the $\hat{f}$ direction and defining ζ as the maximum spread across all window positions.

2.2.3 Branch and bound search tree Equation (5) naturally fits a branch and bound framework with a backtracking search tree, in which each node corresponds to a particular set of samples C (shown in Supplementary Fig. S2). Although this tree exhaustively represents all possible sample sets C, the search is very efficient since most nodes can be pruned from the search tree. First of all, if $V^{(C)}$ becomes equal to the empty set, all child nodes of node C can be discarded because these will also result in the empty set. Second, as a result of the search tree topology, for each node C we can define an upper bound $\hat{f}_{\text{up}}^{(C)}$ and lower bound $\hat{f}_{\text{low}}^{(C)}$. The upper bound $\hat{f}_{\text{up}}^{(C)}$ is defined as the value of $\hat{f}$ that would be obtained assuming all its subnodes do not result in the empty set (best case scenario)

$$\hat{f}_{\text{up}}^{(C)} = \sum_{\tau \in C} w(\tau) + \sum_{\tau \in C_{\text{sub}}} w(\tau), \tag{10}$$

where C_{sub} denotes the collection of all samples in the subnodes of C. The lower bound $\hat{f}_{\text{low}}^{(C)}$ is defined as the value of $\hat{f}$ that would be obtained assuming all subnodes will result in the empty set (worst-case scenario)

$$\hat{f}_{\text{low}}^{(C)} = \sum_{\tau \in C} w(\tau). \tag{11}$$

A vast reduction of the search space is realized by considering the following branch and bound principle: any node C_α can be pruned if there exists a node C_β, for which the following is true:

$$\hat{f}_{\text{up}}^{(C_\alpha)} < \hat{f}_{\text{low}}^{(C_\beta)} - \zeta, \qquad \text{under the condition: } V^{(C_\beta)} \neq \varnothing \tag{12}$$

Thus, if we encountered a branch whose worst-case error is better than the best-case error of another branch, we can safely discard the latter.

After the complete search tree is traversed, the set $V^{(Q)}$ is determined by the union of all the nodes that resulted in a non-empty $V^{(C)}$. In Equation (12), the parameter ζ is included to ensure that set $V^{(Q)}$ includes $\mathbf{n}^*$ (Theorem 2). An optimal leaf ordering is obtained when the samples are sorted based on their weight $w(\tau)$. This ensures that $\hat{f}_{\text{up}}^{(C)}$ decreases as quickly as possible, in effect pruning the tree early in the search. Also, note that most $V^{(C)}$ will contain many duplicates when symmetries in the topology of $\mathcal{B}$ are considered. By filtering these from $V^{(C)}$ before evaluating the succeeding node results in an additional search speedup.

2.2.4 Tolerance level A final, yet influential, search-space reduction is achieved by only considering solutions for which a certain minimum level of association is achieved. This is realized by enforcing that $\hat{f}_{\text{low}}$ can never be below a user defined tolerance level. In other words, for this bounded $\hat{f}_{\text{low}}$, we can write: $\hat{f}'_{\text{low}} = \max(\hat{f}_{\text{tol}}, \hat{f}_{\text{low}})$. As a result, branches for which $\hat{f}_{\text{low}} \leq \hat{f}_{\text{tol}}$ can be pruned even before the search is started. The search procedure is explained by example in Supplementary Figure S2.

2.2.5 Estimating the false discovery rate Because our primary interest lies with the interpretation of the selected genotype markers and combinatorial logic, it is of critical importance to assess frequency of false positives among the networks called significant. Due to the efficiency of the proposed method, it is possible to employ a permutation procedure to obtain a null-distribution for each $\mathcal{B}_j$. From this distribution, it is possible to estimate the false discovery rate (FDR) and the associated q-values by using the method proposed in Storey and Tibshirani (2003). Not surprisingly, in many cases, multiple network topologies yield significant associations with the same gene. The q-values, available for each of the solutions, provide a convenient way of performing selection of the most parsimonious model by accepting only the topology for which the q-value is minimal.

3 RESULTS

3.1 Genetical genomics dataset

The genetical genomics dataset used to demonstrate our method contains genome-wide RNA transcript measurements performed on four related hematopoietic cell populations (Gerrits *et al.*, 2009). These were isolated from the bone marrow of ∼25 BXD recombinant inbred mouse strains that were derived by crossing C57BL/6J (B6) and DBA/2J (D2) (Peirce *et al.*, 2004). A typical analysis of these data includes determining eQTLs, i.e. regions in the genome for which the genotype across strains associates well with RNA transcript levels.

We inferred associations only for the myeloid cell population, as for this cell type data for the largest number ($T = 24$) of unique BXD strains were available. The expression data were preprocessed as described in the Supplementary Methods. Because the CAL networks inferred for highly correlated genes are equivalent, rather than starting the optimization for each gene separately, we constructed gene clusters and searched for CAL networks for the centroids of each gene cluster. To ensure only tightly correlated probes were clustered, we employed a stringent cutoff (correlation distance cutoff 0.2). This resulted in 6139 clusters that were used to determine eQTLs.

Genotype information for the strains was retrieved from The GeneNetwork (http://www.genenetwork.org/dbdoc/BXDGeno.html). Genotype markers that were highly similar across strains and on the same chromosome were also grouped into clusters to prevent the algorithm from finding many combinations of genotype markers that are equivalent (such as the markers in linkage disequilibrium). This resulted in 453 marker clusters ($L = 453$). The cluster centroids were defined as the majority vote of the individual markers in the cluster and were used as putative inputs to the network (see also the Supplementary Methods and Supplementary Figs. S3 and S4).

For setting the tolerance level f_{tol} no straightforward method exists. Preferably, the tolerance level is set close to the final significance threshold to minimize the effort spent on finding optima for gene clusters that can never be significant. We settled for a tolerance level equal to the 75th percentile of the f^{opt} distribution ($f_{\text{tol}} = 7.6$), obtained by computing the f-values associated with each $\mathbf{y}^{\text{opt}}$. Gene clusters for which the maximum f-score is below this tolerance level (i.e. in case $f^{\text{opt}} < f_{\text{tol}}$) were not included in the CAL network search, to result in a set of 1525 high-potential gene clusters.

3.2 Algorithm performance

From the methods section it follows that, under the condition that an appropriate value for ζ is found, our algorithm produces an optimal solution. We empirically validate this claim by comparing solutions of the proposed algorithm with the global optimum obtained with an exhaustive search. To ensure realistic conditions, we do this using the real data described above.

For each gene expression vector, we performed our CAL network search as described with seven network topologies containing AND, OR and XOR logic as well as a more complex combination of these Boolean functions. A rather low tolerance level ($f_{\text{tol}} = 4$) was used, which turned out to capture most of the solution-space (>80% for all topologies). The solutions obtained were compared with the optimal solutions determined by means of an exhaustive search for the same seven Boolean logic functions using Grid computing

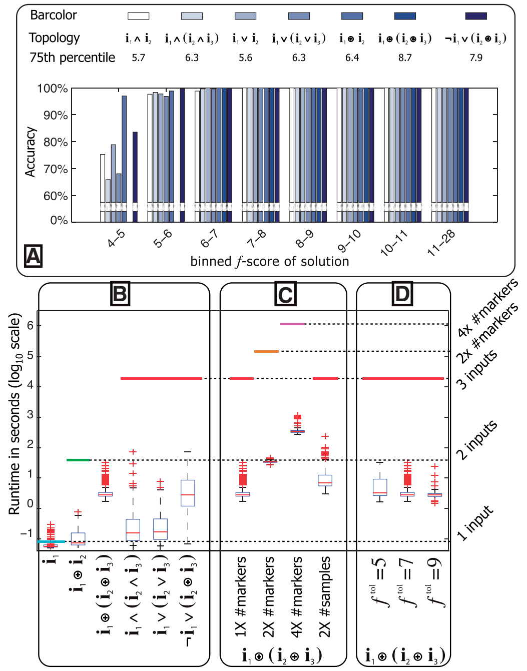


Fig. 4. Algorithm performance in terms of accuracy and runtime under various conditions. (**A**) Bargraph displaying accuracy for different network topologies and different values of the f-score. For each of the network topologies the 75th percentile of the solution distribution is also given, showing that for solutions in the tail 100% accuracy is obtained. For the two missing bars in the 4-6 and 5-6 bins no solutions were found. (**B**) and **C**) Runtimes for different network topologies and dataset sizes. The horizontal lines reflect runtimes for exhaustive search. From bottom to top these represent the runtimes for: a single input network, two input network and three input network with one, two and four times the number of predictors, respectively.

facilities. The accuracy is expressed as the percentage of times that the algorithm finds the same solution as the exhaustive search.

Figure 4A shows the resulting accuracy. We observe that for solutions with f-scores between 5 and 6 already >95% accuracy is achieved, while for solutions with f-scores of $\geq$6, virtually 100% accuracy is achieved for each topology. For comparison, Figure 4A also gives the 75th percentiles of the solution distributions for each topology. Because solutions of interest (putatively significant solutions) are required to have f-scores substantially higher than the 75th percentile, we can conclude that our method achieves 100% accuracy for a reasonable operating range (solutions with f-scores between 4 and 5—where the accuracy is below 95%—are well the 75th percentile for all networks).

While comparing our method to the method presented in Mukherjee *et al.* (2009), using simulated gene expression vectors and a predetermined random network (ground truth), we found that our method reaches higher true positive rates (see Supplementary Material). These results illustrate the benefit of searching for solutions for each of the network topologies separately, and employing a significance estimate to enforce parsimony.

Obtaining the same accuracy as an exhaustive search is only useful if this is achieved for runtimes that are substantially lower. To asses this, we randomly selected 200 gene expression vectors from the 1525 gene clusters and measured runtimes for both our CAL network search as well as the exhaustive search. Figure 4B–D shows these runtimes for a range of conditions. The boxplots represent the results obtained with the CAL network search and the horizontal lines the runtimes for the exhaustive search.

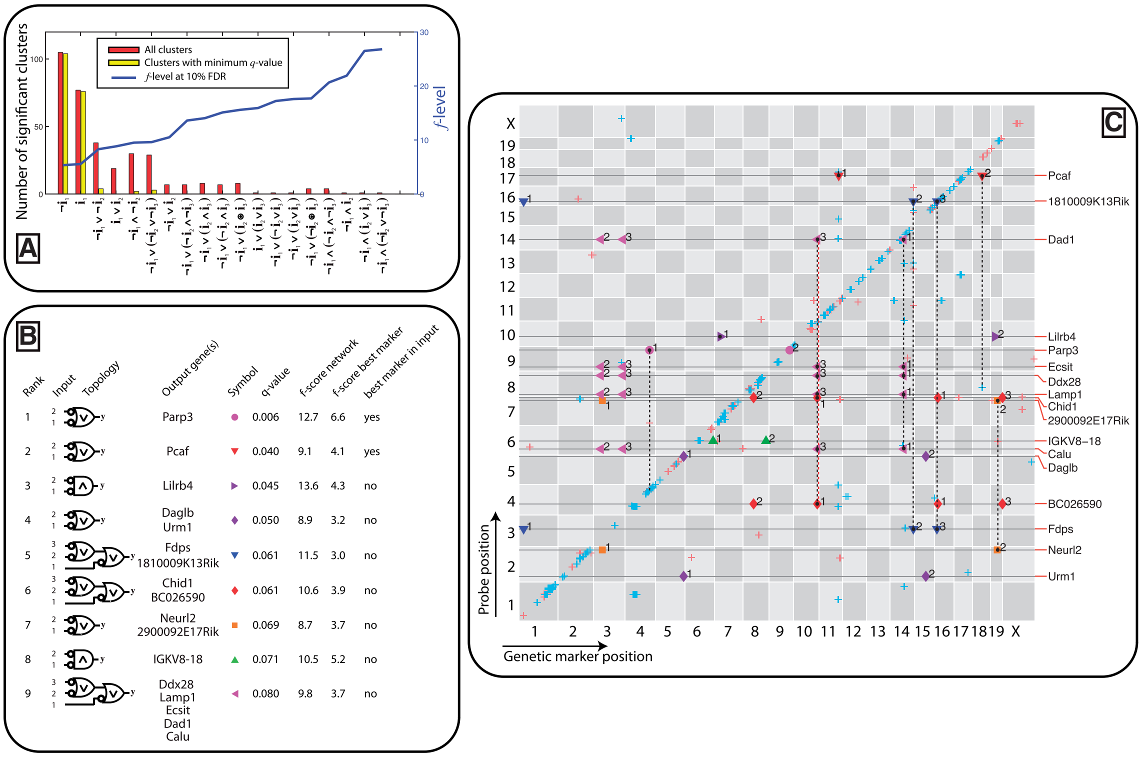
Figure 4B compares runtimes for different network topologies. Clearly, the branch and bound algorithm significantly outperforms the exhaustive search under all experimental conditions with differences in runtime of up to four orders of magnitude. For the three input networks in particular, the runtime required for exhaustive search (>5 h per gene per network) prohibits any further permutation procedures. The CAL network search, on the other hand, is able to find the solution in a matter of seconds, thereby enabling the large number of permutations required to obtain reliable significance estimates.

Compared to the variance in runtime of the exhaustive search, which was negligible, the variance of the CAL network search is quite high. This is expected as our CAL network search finishes rapidly when a good solution presents itself early in the search, while more time is needed to conclude that no acceptable solution is present. For a similar reason, the more complex networks, those containing XOR logic, have higher median runtimes. On no occasion, however, does this increase runtimes >100 s for any of the networks.

To evaluate performance as a function for dataset size we artificially increased the number of predictors and the number of samples (Fig. 4C). In addition, runtimes for different tolerance levels were examined (Fig. 4D). The number of predictors was increased by horizontally concatenating the original matrix L with copies of L containing 10% random bit-flips. The sample size was increased by vertically concatenating matrix L as well as all gene expression vectors $\mathbf{g}$ with copies of L and $\mathbf{g}$, respectively. In case of the latter, normally distributed noise was added to the copies with $\sigma_{\text{noise}} = 0.1\sigma_{\mathbf{g}}$. We observe that for both the exhaustive search as well as the CAL network search runtimes increase substantially as the number of predictors increase. In case of the CAL network search, this is explained by the fact that many very good solutions are present due to the increased imbalance between the number of predictors and the sample size. It is expected, yet not quantitatively established, that better performance is observed when this balance is restored. The increase in runtime as a result of an increased number of samples is moderate, with a median runtime considerably lower than an exhaustive search for only two input networks. Likewise, increasing the tolerance level only moderately speeds up the CAL network search, demonstrating that runtime is robust for the setting of this parameter.

3.3 Combinatorial eQTLs

We performed the CAL network search for the set of 1525 high-potential gene clusters. The complete search (e.g. for all gene clusters and all topologies) was repeated 100 times using a permuted version of the gene-expression vectors. For each topology, this resulted in a null-distribution containing 152 500 values, which was used to estimate q-values for each of the resulting solutions. We considered network topologies with a maximum of three inputs listed in Supplementary Figure S5. Notably, we included two single-input networks to account for direct positive and negative association, respectively, which is equivalent to positive association with the



Rank	Input	Topology	Output gene(s)	Symbol	q-value	f-score network	f-score best marker	best marker in input
1	2 1	∨	Parp3	●	0.006	12.7	6.6	yes
2	2 1	∨	Pcaf	▼	0.040	9.1	4.1	yes
3	2 1	∧	Lilrb4	▶	0.045	13.6	4.3	no
4	2 1	∨	Daglb Urm1	◆	0.050	8.9	3.2	no
5	3 2 1	∨ ∨	Fdps 1810009K13Rik	▼	0.061	11.5	3.0	no
6	3 2 1	∨ ∨	Chid1 BC026590	◆	0.061	10.6	3.9	no
7	2 1	∨	Neurl2 2900092E17Rik	■	0.069	8.7	3.7	no
8	2 1	∧	IGKV8-18	▲	0.071	10.5	5.2	no
9	3 2 1	∨ ∨	Ddx28 Lamp1 Ecsit Dad1 Calu	◀	0.080	9.8	3.7	no

Fig. 5. (**A**) Bargraph with an overview of the number of gene clusters for which a significant (10% FDR) solution is found. Network topologies are sorted according to the 10% FDR level (blue line). (**B**) CAL networks significant at 10% FDR. The color and shape of the symbols correspond to the symbols used in (**C**). Small circles at the inputs of the networks denote negation, i.e. for these inputs the mapping from allele to binary representation is switched. We also indicate whether the best single marker coincides, for that gene cluster, with one of the inputs of the CAL network. (**C**) Marker/probe-plot for the top CAL networks showing both the eQTLs (blue crosses) and ceQTLs (sets of colored symbols of various shapes). The colors and shapes of the markers refer to the network topologies listed in (**B**). Horizontal gray lines connect the inputs and the output of the CAL network. Because probes were clustered, it occurs that the ceQTLs map to multiple probes in case these probes were part of the same cluster. The numeric labels near the the colored symbols correspond to the input of the network. Notably, some probes seem to be predicted by more ceQTLs than there are inputs to the CAL network reported. This occurs when there are multiple combinations of markers that show the same association with the gene expression level of the network output, and can be explained by similarity among markers. The *cis*-band (diagonal) is clearly visible, and in one occasion contains a ceQTL. Overlap among ceQTLs from different networks is marked by red dashed lines, overlap between ceQTLs and eQTLs by black dashed lines.

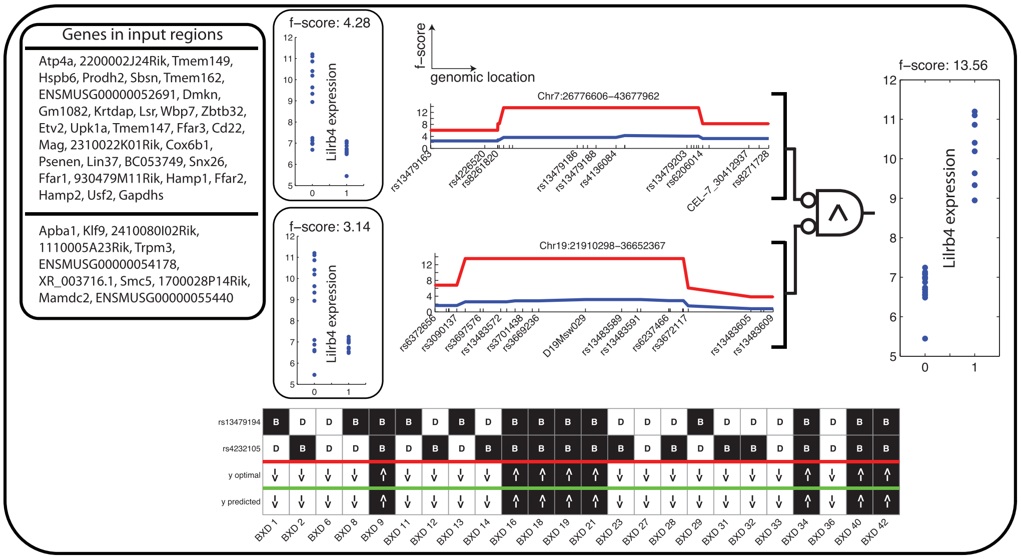
D2 and B6 allele, respectively. This ensures that the algorithm has the option of choosing the least complex model in case an eQTL is capable of explaining a significant portion of the variance in the expression of the gene cluster.

Figure 5A gives an overview of the number of gene clusters for which the output of a CAL network significantly (at the 10% FDR level) associated with its expression (red bars). To obtain additional confidence in the significance threshold, we calculated q-values for 10 additional permutations of the whole dataset. For none of the network topologies did the mean number of significant gene clusters across the 10 permutations exceed 0.6, indicating that the expected number of false discoveries is conservatively kept under control. The yellow bars indicate the number of significant gene clusters after model size selection based on the q-value as detailed in Section 2. It appears that most of the gene clusters for which association is observed can be explained by one of the single input networks. For nine gene clusters (corresponding to 17 genes), however, a CAL network was capable of explaining significantly more of the variance than one of the single input networks or any one of the other CAL networks.

The network topologies, q-values and association scores of the significant CAL networks are given in Figure 5B. Not surprisingly, for all gene clusters at the output of these networks, the *combination* of loci is vastly superior in explaining the variance in expression over any of the markers in isolation. Interestingly, many of these genomic regions would have been missed, as in seven of the networks the best markers do not coincide with one of the inputs of the CAL network.

The sets of markers that were found as the optimal inputs for the seven topologies were mapped onto the genome. Combinatorial eQTLs (ceQTL) were then defined as stretches of consecutive markers. A genome map of the (c)eQTLs is given in Figure 5C, showing the eQTLs (red and blue crosses for positive and negative association, respectively) and ceQTLs (colored symbols) on the x-axis versus the genomic positions of the probes measuring expression on the y-axis. The numbers near the ceQTL symbols correspond to the inputs of the CAL networks depicted in Figure 5B.

Before we zoom in on one of the CAL networks in more detail, some general observations can be made. In particular, we note that in some cases overlap exists among the markers selected at the



Genes in input regions

Atp4a, 2200002J24Rik, Tmem149, Hspb6, Prodh2, Sbsn, Tmem162, ENSMUSG00000052691, Dmkn, Gm1082, Krtdap, Lsr, Wbp7, Zbtb32, Etv2, Upk1a, Tmem147, Ffar3, Cd22, Mag, 2310022K01Rik, Cox6b1, Psenen, Lin37, BC053749, Snx26, Ffar1, 930479M11Rik, Hamp1, Ffar2, Hamp2, Usf2, Gapdhs

Apba1, Klf9, 2410080I02Rik, 1110005A23Rik, Trpm3, ENSMUSG00000054178, XR_003716.1, Smc5, 1700028P14Rik, Mamdc2, ENSMUSG00000055440

	BXD 1	BXD 2	BXD 6	BXD 8	BXD 9	BXD 11	BXD 12	BXD 13	BXD 14	BXD 16	BXD 18	BXD 19	BXD 21	BXD 23	BXD 27	BXD 28	BXD 29	BXD 31	BXD 32	BXD 33	BXD 34	BXD 36	BXD 40	BXD 42
rs13479194	B	D	D	B	B	B	D	B	D	B	B	B	B	D	D	D	B	D	D	D	B	D	B	B
rs4232105	D	B	D	D	B	D	B	D	B	B	B	B	B	B	D	B	D	B	B	D	B	D	B	B
y optimal	↓	↓	↓	↓	↑	↓	↓	↓	↓	↑	↑	↑	↑	↓	↓	↓	↓	↓	↓	↓	↑	↓	↑	↑
y predicted	↓	↓	↓	↓	↑	↓	↓	↓	↓	↑	↑	↑	↑	↓	↓	↓	↓	↓	↓	↓	↑	↓	↑	↑

Fig. 6. Input regions of the CAL network for *Lilrb4* The line graphs give the f-score for association between the output gene and the individual markers (blue) and the network output (red). The latter was computed by taking the maximum f-score of the network using the marker under evaluation for one input and any of the other markers for the second input of the network. Where possible the IDs of the genetic markers are given, but some were omitted for readability. The dot plots gives the expression values separated by network output (right) and the best markers in the inputs (left). Finally, for one particular combination of markers the genotype for all strains is depicted as a Boolean heat map. In these diagrams, the NOT gates were already incorporated.

inputs of the CAL networks and between other network inputs and eQTLs. In seven instances, the identified ceQTLs coincide with eQTLs (connected by black dashed lines in the figure). Some of these eQTLs are located in *cis*. The finding of CAL networks that share one of their inputs (ceQTLs) with an eQTL suggests that the local genotype associated with the eQTL is involved in the regulation of a local gene (*cis*-regulation), but in addition collaborates with the other CAL input locus/loci to regulate the CAL network output gene(s). Furthermore, two of the CAL networks (ranked sixth and ninth) share a ceQTL between the inputs (connected by red dashed lines). It is not inconceivable that a gene present in this ceQTL is indeed involved in the regulation of the target genes of both networks, but that the interaction partners through which this regulation is established differs for both target genes.

Among the list of output genes of the nine most significant CAL networks is *Lilrb4* (ranked third). *Lilrb4* encodes a leukocyte immunoglobulin-like receptor which is expressed on the surface of mast cells, neutrophils, and macrophages. It plays a key role in counter-regulating the inflammatory response to prevent pathologic excessive inflammation (reviewed in Katz, 2007).

Figure 6 shows small regions around the ceQTLs that were selected as inputs for the CAL network of *Lilrb4*. For each region, the association was measured between the expression of *Lilrb4* and the individual markers (blue lines). The red lines, on the other hand, give the association score for the network output. Clearly, the association between the logical combination of inputs and the expression of *Lilrb4* is markedly higher than considering any of the markers in isolation. The regions for which the red curves reach their maximum correspond to the ceQTLs.

The Boolean heat map, displayed at the bottom of Figure 6, outlines the genotype of one particular combination of genetic markers in the ceQTLs across the BXD mouse strains. The bottom two rows of this heat map give the optimal network output and predicted output, respectively. For the *Lilrb4* network the optimal network output is exactly recapitulated by the CAL network. For *Lilrb4* elevated expression is exclusively observed in case of B6 alleles in both the ceQTL regions of Chromosomes 7 and 19.

To focus our attention to the most interesting genes in the ceQTLs we performed a literature search using Ingenuity pathway analysis (Ingenuity©Systems, www.ingenuity.com). Interestingly, we found a substantial number of interactions between genes localized in the ceQTLs and *Lilrb4*. For example, the literature search revealed a link between *Apba1* (located in the ceQTL region on Chromosome 19) and *Lilrb4*. Both protein products have been described to bind ITGB3 (Calderwood *et al.*, 2003; Castells *et al.*, 2001). In addition, the search revealed a link between *Psenen* (Chromosome 7 ceQTL) and *Apba1* (Chromosome 19 ceQTL). Both protein products have been described to bind PSEN1 and PSEN2 (Biederer *et al.*, 2002; Steiner *et al.*, 2002).

While literature is able to link the genes in the ceQTLs to *Lilrb4* and thereby gives the first clues as to how the expression of *Lilrb4* may be regulated, we do not exclude that other interactions (not yet represented in literature) exist. In any case, the result of our method should provide a set of testable hypotheses that can be validated in the laboratory.

4 DISCUSSION

Unravelling (transcriptional) regulatory networks by inferring complex associations, for instance, between genotype and gene expression, necessitates algorithms that take into account possible (allele-specific) interactions. For this purpose, we have proposed a method to efficiently infer CAL networks, i.e. small logic networks in which allele-specific interactions are modeled by Boolean functions. To find the best possible fit of the model given the data, a computationally challenging optimization problem had to be solved.

This was achieved by rewriting the optimization such that it could be effectively solved by a customized branch and bound algorithm. Proof and empirical evidence for optimality of the solution, under appropriate conditions, was given. At the same time, differences in runtimes of up to four orders of magnitude were observed when compared to exhaustive search.

Because the CAL network search is able to find the optimal solution in a matter of seconds a permutation procedure becomes feasible, which can be employed to obtain estimates of the FDR. This is a major advantage as the resulting q-values allow selection of the most parsimonious model and enable ranking the network topologies in terms of their complexity.

We demonstrated our algorithm on a genetical genomics dataset, and found that, from the 1525 gene clusters (2913 genes) that resulted after selection of high potential genes, 9 gene clusters (17 genes) were significantly associated (at 10% FDR level) through a logical combination of genomic loci rather than a single eQTL. Notably, without incorporating the complex interactions, these associations would have gone unnoticed. Many of the discovered input regions were found to overlap eQTLs or were shared inputs of CAL networks explaining the expression of other genes, suggesting that these regions, indeed, are involved in transcriptional regulation.

ACKNOWLEDGMENTS

The authors thank Daoud Sie and Leonid V. Bystrykh for their valuable input.

Funding: This work was part of the BioRange programme of the Netherlands Bioinformatics Centre (NBIC), which is supported by a BSIK grant through the Netherlands Genomics Initiative (NGI); Dutch Life Science Grid initiative of NBIC and the Dutch e-Science Grid BiG Grid, SARA - High Performance Computing and Visualisation; Netherlands Genomics Initiative (Horizon, 050-71-055); Dutch Cancer Society (RUG2007-3729); Netherlands Organization for Scientific Research (NWO) (VICI to G.d.H., 918-76-601); European Community (EuroSystem, 200720).

Conflict of Interest: none declared.

REFERENCES

Biederer,T. *et al.* (2002) Regulation of APP-dependent transcription complexes by mint/x11s: differential functions of Mint isoforms. *J. Neurosci.*, **22**, 7340–7351.

Bystrykh,L.V. *et al.* (2005) Uncovering regulatory pathways that affect hematopoietic stem cell function using 'genetical genomics'. *Nat. Genet.*, **37**, 225–232.

Calderwood,D.A. *et al.* (2003) Integrin beta cytoplasmic domain interactions with phosphotyrosine-binding domains: a structural prototype for diversity in integrin signaling. *Proc. Natl Acad. Sci. USA,* **100**, 2272–2277.

Castells,M.C. *et al.* (2001) gp49b1-alpha(v)beta3 interaction inhibits antigen-induced mast cell activation. *Nat. Immunol.*, **2**, 436–442.

Cleveland,W. (1979) Robust locally weighted regression and smoothing scatterplots. *J. Am. Stat. Assoc.*, **74**, 829–836.

Evans,D.M. *et al.* (2006) Two-stage two-locus models in genome-wide association. *PLoS Genet.*, **2**, e157.

Frankel,W.N. and Schork,N.J. (1996) Who's afraid of epistasis? *Nat. Genet.*, **14**, 371–373.

Gerrits,A. *et al.* (2009) Expression quantitative trait loci are highly sensitive to cellular differentiation state. *PLoS Genet.*, **5**, e1000692.

Jansen,R.C. and Nap,J.P. (2001) Genetical genomics: the added value from segregation. *Trends Genet.*, **17**, 388–391.

Katz,H.R. (2007) Inhibition of pathologic inflammation by leukocyte Ig-like receptor B4 and related inhibitory receptors. *Immunol. Rev.*, **217**, 222–230.

Kooperberg,C. and Ruczinski,I. (2004) Identifying interacting SNPs using Monte Carlo logic regression. *Genet. Epidemiol.*, **28**, 157–170.

Ljungberg,K. *et al.* (2004) Simultaneous search for multiple QTL using the global optimization algorithm direct. *Bioinformatics*, **20**, 1887–1895.

Manichaikul,A. *et al.* (2009) A model selection approach for the identification of quantitative trait loci in experimental crosses, allowing epistasis. *Genetics*, **181**, 1077–1086.

Michaelson,J.J. *et al.* (2009) Detection and interpretation of expression quantitative trait loci (eQTL). *Methods*, **48**, 265–276.

Mukherjee,S. *et al.* (2009) Sparse combinatorial inference with an application in cancer biology. *Bioinformatics*, **25**, 265–271.

Nunkesser,R. *et al.* (2007) Detecting high-order interactions of single nucleotide polymorphisms using genetic programming. *Bioinformatics*, **23**, 3280–3288.

Peirce,J.L. *et al.* (2004) A new set of bxd recombinant inbred lines from advanced intercross populations in mice. *BMC Genet.*, **5**, 7.

Pollack,J.R. *et al.* (2002) Microarray analysis reveals a major direct role of DNA copy number alteration in the transcriptional program of human breast tumors. *Proc. Natl Acad. Sci. USA*, **99**, 12963–12968.

Schadt,E.E. *et al.* (2003) Genetics of gene expression surveyed in maize, mouse and man. *Nature*, **422**, 297–302.

Shames,D.S. *et al.* (2006) A genome-wide screen for promoter methylation in lung cancer identifies novel methylation markers for multiple malignancies. *PLoS Med.*, **3**, e486.

Steiner,H. *et al.* (2002) Pen-2 is an integral component of the gamma-secretase complex required for coordinated expression of presenilin and nicastrin. *J. Biol. Chem.*, **277**, 39062–39065.

Storey,J.D. and Tibshirani,R. (2003) Statistical significance for genomewide studies. *Proc. Natl Acad. Sci. USA*, **100**, 9440–9445.

Visel,A. *et al.* (2009) Chip-seq accurately predicts tissue-specific activity of enhancers. *Nature*, **457**, 854–858.

Wongseree,W. *et al.* (2009) Detecting purely epistatic multi-locus interactions by an omnibus permutation test on ensembles of two-locus analyses. *BMC Bioinformatics*, **10**, 294.

Zhang,Y. and Liu,J.S. (2007) Bayesian inference of epistatic interactions in case-control studies. *Nat. Genet.*, **39**, 1167–1173.