Genetic Epidemiology 28: 157–170 (2005)

Identifying Interacting SNPs Using Monte Carlo Logic Regression

Charles Kooperberg[1]* and Ingo Ruczinski[2]

[1]*Division of Public Health Services, Fred Hutchinson Cancer Research Center, Seattle, Washington*
[2]*Department of Biostatistics, Bloomberg School of Public Health, Johns Hopkins University, Baltimore, Maryland*

Interactions are frequently at the center of interest in single-nucleotide polymorphism (SNP) association studies. When interacting SNPs are in the same gene or in genes that are close in sequence, such interactions may suggest which haplotypes are associated with a disease. Interactions between unrelated SNPs may suggest genetic pathways. Unfortunately, data sets are often still too small to definitively determine whether interactions between SNPs occur. Also, competing sets of interactions could often be of equal interest. Here we propose Monte Carlo logic regression, an exploratory tool that combines Markov chain Monte Carlo and logic regression, an adaptive regression methodology that attempts to construct predictors as Boolean combinations of binary covariates such as SNPs. The goal of Monte Carlo logic regression is to generate a collection of (interactions of) SNPs that may be associated with a disease outcome, and that warrant further investigation. As such, the models that are fitted in the Markov chain are not combined into a single model, as is often done in Bayesian model averaging procedures. Instead, the most frequently occurring patterns in these models are tabulated. The method is applied to a study of heart disease with 779 participants and 89 SNPs. A simulation study is carried out to investigate the performance of the Monte Carlo logic regression approach. *Genet. Epidemiol.* 28:157–170, 2005.

Key words: association studies; binary variables; Boolean logic; haplotype

Grant sponsor: NIH; Grant number: CA 074841, CA 053996, HL 74745; Grant sponsor: Maryland Cigarette Restitution Fund.
*Correspondence to: Charles Kooperberg, Division of Public Health Sciences, Fred Hutchinson Cancer Research Center, PO Box 19024, Seattle, WA 98109-1024. E-mail: clk@fhcrc.org
Received 30 December 2003; Revised 28 May 2004; Accepted 7 September 2004
Published online 5 November 2004 in Wiley InterScience (www.interscience.wiley.com)
DOI: 10.1002/gepi.20042

INTRODUCTION

With many advances in high-throughput sequencing techniques, an increasingly large number of studies are being carried out to associate clinical outcomes with single-nucleotide polymorphisms (SNPs). How to best conduct such a study and how to analyze the data most efficiently remain open questions.

In this paper, we are concerned with the situation where we have data on a large number of genes, with only a few SNPs on each gene. Those SNPs may have been selected because of previous research, or maybe as a set of tag-SNPs [e.g., Johnson et al., 2001; Weiss and Clark, 2002; Sebastiani et al., 2003] that jointly are sufficient to identify the common haplotypes. While those tag-SNPs could be used to estimate the haplotypes in all participants, it is reasonable to use individual SNPs in an association study when only a few of those on each gene are available.

No matter what the exact study design is, when association studies with multiple SNPs are carried out, determining whether any interactions between those SNPs are associated with the outcome will be of great interest. In the situation where individual SNPs on multiple genes interact, this may suggest biological pathways; when SNPs on the same gene interact, these interactions may effectively group the haplotypes in those that are and those that are not associated with the outcome.

Recently a few methods were proposed for directly finding interactions between SNPs. Nelson et al. [2001] proposed a combinatorial partitioning method to identify combinations of SNPs that predict a quantitative outcome. In their approach, a limited number of SNPs is selected among those at which sequence information is available. All combinations of those selected SNPs are further considered. Various combinatorial arguments make it possible to apply this approach to fairly sizable data sets. Ritchie et al. [2001]

extended this methodology and applied it to a breast cancer data set. Both approaches tried to identify combinations of SNPs that are associated with disease status, without being concerned with generating a parsimonious, interpretable rule. It is not clear whether these techniques could be applied to data with hundreds of SNPs, as likely will be collected in the near future. Zee et al. [2002] also first selected a smaller number of SNPs among those that are available. They then used a logistic regression model with stepwise model selection. We discuss this approach below in Comparison to Stepwise Selection, as they analyzed the same data as we do in this paper. Hoh and Ott [2003] provided an overview of multilocus approaches to localizing complex human trait genes. Their paper discussed the abovementioned techniques as well as related approaches using family data or haplotype data.

In most published studies, few interactions between SNPs were identified. There may be several reasons for this. While high-throughput sequencing technologies have made it much easier to collect a large number of SNPs, many of the data sets analyzed are still relatively small, and combined with a multiple comparisons correction of significance tests, this implies that there is limited power to identify interactions with moderate effects. In such a situation, perhaps the best strategy is to identify a small number of combinations of SNPs that are potentially associated with an outcome. Ideally, in other studies, hypotheses based on these results can be further investigated.

Such a hypothesis-generating analysis is different from a confirmatory analysis. Putting it simply, we are not interested in identifying a single model (that is significant at, say, the 5% level), but we want to identify a larger number of alternative models. Each of these models individually may not stand up to a rigorous 5% significance cutoff, but jointly there may be strong evidence that there is an association. Such a limited number of associations can then be examined using other data sources.

Logic regression [Ruczinski et al., 2003] is a generalized regression method that is intended for situations where most covariates are binary. It was successfully applied to SNP data [Kooperberg et al., 2001]. As designed, logic regression is a methodology that tries to determine a single model which may involve various combinations of SNPs that are associated with a clinical outcome. Because of the rigor with which model selection is carried out to correct for multiple comparisons, in situations where the true effects are small, the power to identify higher-order interactions is limited.

In this paper, we describe a new methodology, Monte Carlo logic regression. This methodology combines logic regression and Markov chain Monte Carlo (MCMC) model selection to identify a larger group of SNPs that are potentially associated with a clinical outcome. As opposed to the methods described above, including logic regression, the goal in Monte Carlo logic regression is not to identify a single ''best'' model that relates SNPs to the clinical outcome of interest. Instead, the new approach explores a large number of potential logic regression models using an MCMC mechanism. The idea is that those models that are identified frequently during the MCMC iterations are good candidates for follow-up studies. We apply the method to data on a study of heart disease [Hoh et al., 2001; Zee et al., 2002] where 89 SNPs in 62 candidate genes were genotyped for all 779 heart disease patients. A simulation study illustrates some of the strengths and weaknesses of the proposed approach.

BACKGROUND: LOGIC REGRESSION FOR SNP DATA

Logic regression is a generalized regression method that is intended for situations where most covariates are binary, such as is the case for SNP data. Logic regression is described in detail in Ruczinski et al. [2003]. Monte Carlo logic regression, which is introduced in this paper, uses logic regression models. However, both the selection of those models, and the interpretation of the results, are very different for both approaches.

Let Y be a phenotype trait which can be either binary (e.g., diseased vs. nondiseased) or quantitative (e.g., blood pressure). The logic regression model is

$$g[E(Y|\mathbf{X})] = \beta_0 + \sum_{i=1}^{K} \beta_i L_i(\mathbf{X}) \qquad (1)$$

where g is an appropriate link function, $\mathbf{X}$ are the covariates, $\beta_0, \ldots, \beta_K$ are parameters, and the $L_i(\mathbf{X})$ are Boolean combinations of the covariates, such as $X_1^c \wedge (X_2 \vee X_3)$. We refer to the L_i as a logic tree, as the L_i are organized in a tree form; see Figure 1. Using this ''logic tree' representation, it is possible to obtain any other logic tree by a finite

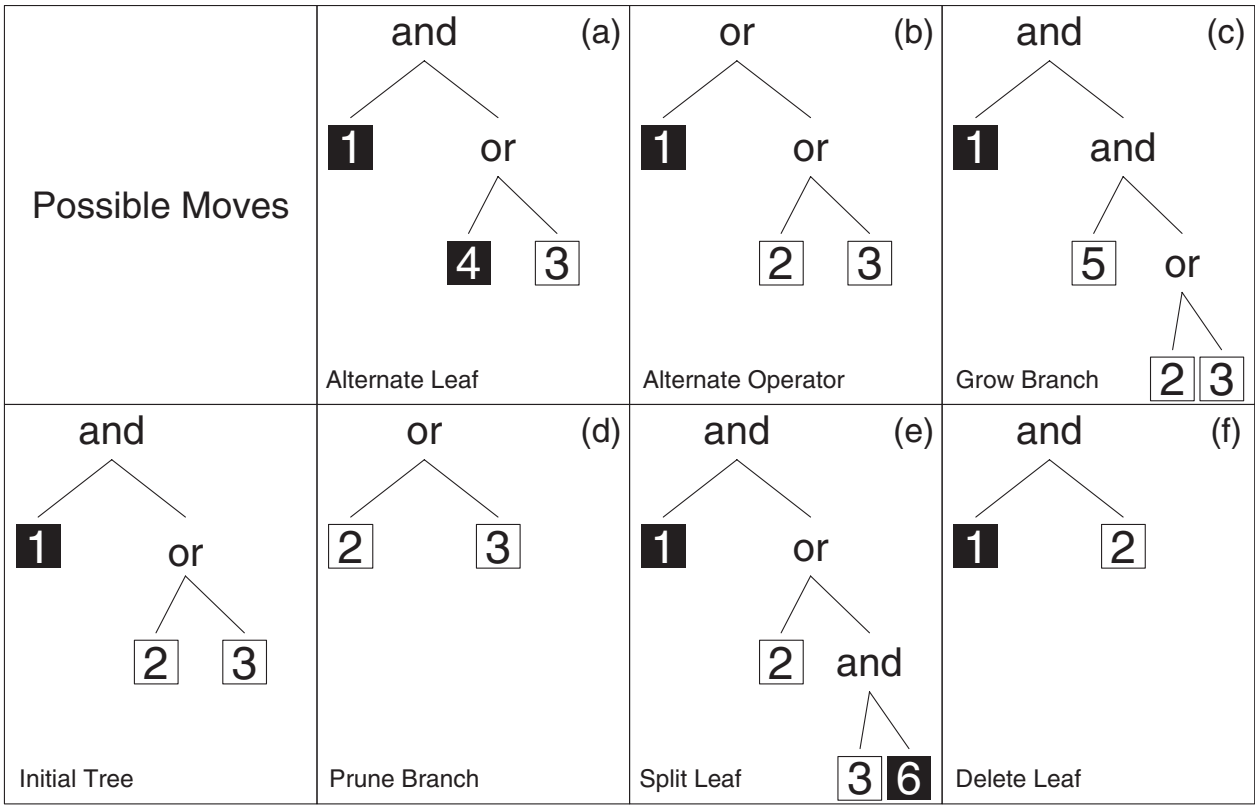

Fig. 1. Logic tree representation of $X_1^c \wedge (X_2 \vee X_3)$, and permissible moves for logic trees. We use white numerals on black background to indicate complement of a covariate. Starting tree is at lower left. Moves are illustrated in a–f.

number of operations such as growing of branches, pruning of branches, and changing of leaves; see Figure 1.

In regular logic regression, these logic trees are selected adaptively, using a simulated annealing algorithm. We start with $L=0$. Then, at each stage a new tree is selected at random among those that can be obtained by simple operations on the current tree. This new tree always replaces the current tree if it has a better score than the old tree, and otherwise is accepted with a probability that depends on the difference between the scores of the old and the new tree, and the stage of the algorithm. Early on, trees with considerably worse scores are still accepted, while toward the end of the algorithm, the probability of accepting a tree with a worse score becomes eventually almost zero. In this simulated annealing algorithm, each covariate could end up in multiple trees. Note that the dimensionality of the model (1) is not the number of covariates, which may be very large, but the number of parameters, which is the number of logic trees plus one, and is usually small.

As the best fitting model typically overfits the data, model selection is of critical importance. Logic regression model selection is carried out using cross-validation. For cross-validation, the data are repeatedly split in a training and test set. Logic regression models with various numbers of logic trees and various numbers of terms in those trees are fit on the training data. The model size which has the best score on the test data is then selected, and a model of that size is computed on the complete data. Alternatively, a set of randomization tests can be used to avoid this overfitting problem [Ruczinski et al., 2003].

For SNP data, one could let each individual SNP take values 0, 1, and 2 for the number of variant alleles at the respective SNP. Alternatively, we propose to code an SNP S_i into two binary covariates: $X_{i1}=1$ if the SNP has at least one variant allele, and $X_{i2}=1$ if it has two variant alleles; both are 0 otherwise. One can see that X_{i1} codes the dominant effect of SNP S_i, and X_{i2} codes the recessive effect of SNP S_i. This coding has the advantage that each x_{ij}, $j=1, 2$, corresponds directly to the mode of the genetic effect of SNP

S_i. We applied logic regression successfully to the simulated SNP data of the Twelfth Genetic Analysis Workshop and a study of heart disease using this coding [Kooperberg et al., 2001; Ruczinski et al., 2004]. In the simulated data set, all mutations were correctly identified without any false positives.

Two limitations of logic regression are that: (1) logic regression identifies a single best model as in equation (1), while in practice there may be alternative models that fit the data almost equally well; and (2) if a covariate X is highly correlated with a covariate that is selected by the simulated annealing algorithm, then X will likely not end up in the model, while in practice the data at hand may not be able to distinguish between the two ''competing'' covariates. This is an important issue in analyzing SNP data, as SNPs are frequently observed in linkage disequilibrium. Below, we describe an alternative methodology to address these problems.

METHODS: MONTE CARLO LOGIC REGRESSION

The goal of Monte Carlo logic regression is to identify all models and combinations of covariates that are potentially associated with the outcome, rather than to construct a single model to predict the outcome. We adopt Bayesian model selection techniques, using Markov chain Monte Carlo (MCMC) to explore a large number of good-fitting models. Unlike many Bayesian model selection problems, where the models that are visited in an MCMC run are averaged to construct predictors that are better than individual covariates, we construct summary measures describing features of all models that were visited.

Specifically, our implementation uses the reversible jump MCMC algorithm of Green [1995]. We select a prior on the model size and a prior on all logic regression models of a given size. The size of the model is defined as $\sum_{i=1}^{K} |L_i|$, where $|L_i|$ is the number of terminal nodes in the logic tree L_i. For example, the model $\beta_0 + \beta_1(X_1 \vee (X_7 \wedge X_{13}^c)) + \beta_2 X_3$ has size four. In Hansen and Kooperberg [2002], it was argued that for polynomial spline routines, a geometric prior on model size has the effect of an Akaike Information Criterion (AIC)-like penalty of the form $-2 \times$ log-likelihood$+\alpha \times$ (number of parameters). This is no longer true for logic regression, as the number of parameters of the model does not need to increase with model size. (Note that in the example above, the model has three parameters.) From a practical viewpoint, however, using a geometric prior still makes sense: models with a larger size typically overfit the data, and thus smaller models are preferred.

The prior on model size does influence the total number of SNPs selected, and thus in a Bayesian sense the probability that a SNP is associated with the outcome depends on the choice of the prior. However, as we will see in our analysis, the actual choice of the prior on the model size has little influence on which SNPs occur most often in the models that are selected; and the relative ordering of the SNPs is what drives most of our analysis.

We use a uniform prior on all logic regression models of a particular size. This requires us to count the number of possible logic regression models of a particular size. As we are less interested in the coefficients β_i than in the selected logic trees L_i in equation (1), we choose not to put a prior on the coefficients β_i, but rather use the method of maximum likelihood to estimate them. Formally this makes our Monte Carlo logic regression algorithm not a Bayesian logic regression algorithm.

Having selected priors, a reversible-jump MCMC algorithm is a modification of the simulated annealing algorithm of Ruczinski et al. [2003]. That is, at each stage of the algorithm we select one of the logic trees at random to modify, and for that tree we select one of the moves that are shown in Figure 1, according to prespecified selection probabilities depending on which of these moves are in fact possible. Some of these moves require the selection of other quantities, such as which specific covariate will enter the logic tree. Note that not all types of moves may be possible at each time. After a new model is selected, we compute the likelihood ratio, the prior ratio, and the posterior ratio [Green, 1995]. Details of implementing a reversible-jump algorithm in a similar problem can be found in Denison et al. [1998]. A few more details of our algorithm can be found in the Appendix. Our algorithm is implemented in Fortran and C with an interface to R and S-Plus and is publically available.

Burn-in and mixing of the MCMC algorithm can be examined by monitoring summary statistics such as the size of the model land when certain particularly simple models (e.g., the null model and models consisting of one logic tree with one particular covariate) are visited. In all our

examples, these statistics suggest that a short burn-in is sufficient, and that chains mix well. However, because of the size of the model space, a long chain after burn-in may still be required.

IDENTIFYING INTERESTING FEATURES

The goal for Monte Carlo logic regression is not to identify a single best model, but to identify potential factors that are associated with an outcome. Different Boolean expressions may be logically identical, and there are no universal algorithms to reduce Boolean expressions. The effect of this is that the number of logic regression models that are visited by the Monte Carlo logic regression algorithm can be huge. Thus, instead of summarizing all those models, we gather some simple statistics about their size, and how often individual SNPs and certain combinations of SNPs occur in them.

- Distribution of size of models. Typically the average model size of the models visited by the Monte Carlo logic regression algorithm will be larger than the average of the prior distribution for the size of the model, even if there is no signal in the data. Thus, rather than to compare the average model size to the prior model size, a better comparison for the average model size is the average model size on a data set in which the response has been randomly permuted. This is just a common randomization test: if the average model size of the Monte Carlo logic regression algorithm applied on the real data after burn-in is larger than the average model size of the same algorithm applied to the permuted data, the covariates are likely jointly related to the outcome.
- Fraction of models that contain a particular covariate (SNP). Covariates may interact with other covariates in their association with the outcome, but by themselves have little marginal association. The fraction p_i of the models that include a covariate X_i is a direct measure of the importance of this covariate for predicting the outcome, rather than just its marginal association.
- Fraction p_{ij} of models that include a pair of SNPs, X_i and X_j, in the same logic regression tree. This quantity summarizes whether an interaction between two covariates may be associated with the outcome of interest. If the two SNPs are in the same logic regression model, but not in the same logic tree, this suggests no interaction.
- Similarly, we can count how often triplets, quadruplets, and so on, of covariates occur jointly in logic trees.

RESULTS

We apply Monte Carlo logic regression to data from a study of heart disease [Hoh et al., 2001; Zee et al., 2002]. Among 779 heart disease patients, 342 showed restenosis (a renarrowing or blockage of an artery at the same site where treatment has already taken place) 6 months after angioplasty, while the other 437 did not. All individuals were genotyped for 89 SNPs in 62 candidate genes that were previously associated with heart disease. Each SNP is coded by two binary covariates, one for the dominant and one for the recessive effect of the SNP. After removing covariates with no variation, this yielded a total of 169 binary covariates. As for most genes there are only one or two SNPs, these data do not exhibit much linkage disequilibrium (except for one gene noted below), and we focus our discussion on the identification of which SNPs are associated with restenosis. We will discuss a situation with linkage disequilibrium in the simulation study.

In Ruczinski et al. [2004], the same data were analyzed using logic regression, considering models with at most three logic trees and up to eight terms. Based on model selection using cross-validation and randomization tests, a model with one logic tree and four terms was selected:

$$(\mathrm{TP53(P72R)}_d^c \wedge \mathrm{CBS(I278T)}_r^c) \vee \mathrm{CD14}_d \vee \mathrm{ADRB3}_r \tag{2}$$

where the subscripts d and r refer to the dominant and recessive coding of the particular SNP, respectively. It was also concluded that beyond the inclusion of $\mathrm{TP53(P72R)}_d$ and $\mathrm{CD14}_d$ in the model, the selection of a single model was not clear-cut, and that various other SNPs may be associated with restenosis as well.

MONTE CARLO LOGIC REGRESSION

We ran the Monte Carlo logic regression algorithm using models with at most K=1, 2, 3, 4 logic trees and a prior on the model size S with $P(S{=}i) \propto a^i$, i=0, 1, 2,…, and with the parameter $a{=}1/\sqrt{2}$, 1/2, and 1/3. Note that $E(S){=}a/(1{-}a)$. For each of these 12 combinations of K and a, we ran three MCMC chains of 5,000,000 models, after a burn-in of 10,000 iterations. (One such chain

took about 15 min of CPU time on one Intel 2.4 GHz processor.) We also ran chains of 5,000,000 models and 10,000 burn-in iterations, using 25 data sets for which the response was randomly permuted for the same choices of K and a. Diagnostic plots suggested that this burn-in was more than sufficient, as some of the more common models (e.g., the null model) are regularly visited during the burn-in and throughout the iterations. Monitoring of the model size also allowed us to conclude that the chains never got stuck.

Table I shows the posterior model size over the all chains for the real data and the permuted data after burn-in together with the prior model size. One can see that the average posterior model sizes for the real data are substantially larger than the corresponding ones for the permuted data, suggesting that there is signal in the data. The difference between the posterior model size for the permuted data and the prior model size gives some indication of the amount of overfitting in the model selection. Thus, this implies that half or more of the signal in the posterior model for the real data may be due to overfitting. As the posterior model sizes for the real data for models with up to four trees are only slightly larger than those for models with up to three trees for each a, we conclude that using at most K=3 logic trees is probably sufficient to model the data.

Table II shows the fraction p_i for the top 15 SNPs that occur most frequently in the models after burn-in. For models with at most K=3 trees and $a=1/\sqrt{2}$ and models with at most K=2 trees and a=1/3, the top 15 SNPs are actually the same, although the ordering differs slightly. There was a large amount of agreement between all (a, K) combinations, as at least 13 of these 15 SNPs in Table II were among the top 15 SNPs for other combinations of (a, K).

One SNP (TNFR1) has both its recessive and dominant coded variables among the top 15 SNPs in Table II. The CBS gene has two different SNPs in Table II. In fact, these two SNPs are in close linkage disequilibrium, and differ only for 3 of the 779 people in the study. The four SNPs in the regular logic regression model (2) are all in Table II.

Table III shows the top seven pairs of SNPs X_i and X_j for which the fraction p_{ij} of the models that include both SNPs *in the same logic tree* was the highest, for the same two (a, K) combinations as in Table II. These fractions give an indication whether an interaction between two SNPs is associated with the outcome. As two SNPs that are frequently in models would be expected to occur more often together in the same logic tree by chance, we compare the observed fraction by an estimate of the expected fraction $\tilde{p}_{ij}$ of times that these SNPs would occur together if SNPs were selected independently with probabilities proportional to p_i and p_j. Set $\tilde{p}_{ij} = \gamma p_i p_j$, where p_i and p_j are the estimates of the marginal posterior probabilities (e.g., as shown in Table II), and γ is a proportionality constant to ensure that $\sum_{ij} p_{ij} = \sum_{ij} \tilde{p}_{ij}$. The magnitude of the ratio $p_{ij}/\tilde{p}_{ij}$

TABLE I. Average posterior model size[a]

		Maximum number of fitted logic trees			
a	Mean of prior	1	2	3	4
Actual data: mean over 3 chains					
$1/\sqrt{2}$	2.41	3.02	5.07	6.34	7.04
1/2	1.00	1.76	2.24	2.48	2.55
1/3	0.50	1.05	1.13	1.24	1.28
Mean over 25 randomizations					
$1/\sqrt{2}$	2.41	2.71	4.37	5.05	5.50
1/2	1.00	1.58	1.91	1.97	2.09
1/3	0.50	0.93	1.03	1.05	1.09

[a]For both actual data and randomizations, SD of realizations is about 10% of mean. For all a and K, mean for actual data is significantly different from randomizations at $P<10^{-5}$, based on a t-test treating each chain and randomization as a single independent observation.

TABLE II. Fraction p_i of models that include particular SNP X_i for top 15 SNPs for which this fraction is largest

	Fraction of times included in model	
SNP	K=3 trees, $a=1/\sqrt{2}$	K=2 trees, a=1/2
TP53(P72R)$_d$[a,b]	0.379	0.200
CD14$_d$[a,b]	0.353	0.201
MDM2$_d$[b]	0.135	0.054
CBS(I278T)$_r$[a,b]	0.132	0.054
TNFR1$_d$[b]	0.119	0.055
CBS(68bp ins)$_r$	0.112	0.046
IL4RA(150V)$_r$	0.110	0.048
TNFR1$_r$[b]	0.105	0.042
APOC3(T3206G)$_d$	0.096	0.038
LTA$_r$	0.076	0.032
GNB$_d$	0.073	0.026
ADRB3$_r$[a]	0.064	0.025
NOS3$_r$[b]	0.063	0.026
LPA(G21A)$_d$	0.055	0.024
ITGB3$_r$	0.053	0.023

[a]SNPs in regular logic regression model of Ruczinski et al. [2004].
[b]SNPs in model of Zee et al. [2002]; their model also included APOC3(C1100T).

TABLE III. Fraction of times that two SNPs are included in same logic tree (p_{ij}, observed), compared to how often those SNPs would be expected to be jointly in model because of marginal fractions ($\tilde{p}_{ij}$, expected) and ratio between these two columns, for seven combinations with largest p_{ij}

		K=3 trees, a=1/$\sqrt{2}$			K=2 trees, a=1.2		
SNP 1	SNP 2	Observed	Expected	Ratio	Observed	Expected	Ratio
TP53(P72R)$_d$	CD14d	0.1816	0.0721	2.52	0.0837	0.0374	2.23
TP53(P72R)$_d$	CBS(I278T)$_r$	0.0770	0.0269	2.85	0.0275	0.0100	2.77
TNFR1$_r$	APOC3(T3206G)$_d$	0.0736	0.0055	13.42	0.0304	0.0015	20.16
CD14$_d$	CBS(I278T)$_r$	0.0612	0.0251	2.43	0.0235	0.0100	2.34
TP53(P72R)$_d$	CBS(68bp ins)$_r$	0.0610	0.0229	2.67	0.0217	0.0085	2.55
CD14$_d$	CBS(68bp ins)$_r$	0.0469	0.0213	1.60	0.0177	0.0086	1.26
TP53(P72R)$_d$	MDM2$_d$	0.0444	0.0277	1.60	0.0128	0.0101	1.26

suggests the extent to which an interaction between SNPs X_i and X_j is present.

We note from Table III that, as expected, the two SNPs with the highest p_i, TP53(P72R)$_d$ and CD14$_d$, appear most often jointly in a logic tree. However, the third most occurring pair of SNPs, TNFR1$_r$ and APOC3(T3206G)$_d$, were only the eighth and ninth SNPs in the marginal ordering. Thus, an interaction between these two SNPs would be a good target for further investigation.

If two SNPs are highly correlated, an interaction between those two SNPs will occur much less often than would be expected by chance, because when one SNP is in the model, the second SNP will add little information, and our model selection strategy will prefer a smaller model without the second SNP. An example is the combination of CBS(I278T)$_r$ and CBS(68bp ins)$_r$. These two SNPs, which differ only in 3 of 779 people in the study, would be expected to be jointly in a logic tree $\tilde{p}_{ij} = 0.80\%$ of the time. However, they only occur together p_{ij}=0.17% of the time for a=1/$\sqrt{2}$ and K=3. Other than the two CBS SNPs, there is little linkage disequilibrium in our data, as most genes have only one or two SNPs. One of the models in our simulation study (model 1) was specifically created to study how Monte Carlo logic regression works when SNPs are highly correlated.

Interactions among triplets of SNPs can be judged similarly. However, no ''expected'' fraction combining the univariate fractions p_i and the pairwise fractions p_{ij} exists, as there is no simple ''trivariate independence'' model based on univariate and bivariate frequencies, other than complete independence, which is no longer appropriate if the covariates are not pairwise independent. The two most frequently occurring triplets are (TP53(P72R)$_d$, CD14$_d$, CBS(I278T)$_r$) and (TP53(P72R)$_d$, CD14$_d$, CBS(68bp ins)$_r$), which occur 2–3 times more than the next triplet; see Table IV. As the two CBS SNPs are identical for almost all people, these two triplets are in fact almost identical as well. Thus, a three-way interaction between CD14, TP53(P72R), and the CBS gene would appear to be a worthwhile target for further investigations.

TABLE IV. Fraction of times that three SNPs are together in same logic tree for three combinations with largest fraction

SNP 1	SNP 2	SNP 3	K=3 trees, a=1/$\sqrt{2}$ Observed	K=2 trees, a=1/2 Observed
TP53(P72R)$_d$	CD14$_d$	CBS(I278T)$_r$	0.0581	0.0223
TP53(P72R)$_d$	CD14$_d$	CBS(68bp ins)$_r$	0.0439	0.0167
TP53(P72R)$_d$	CD14$_d$	APOC3(T3206G)$_d$	0.0204	0.0073

Typically the result of a Monte Carlo logic regression simulation model will not be a single model, but rather a collection of tables like Tables I–IV. Instead, if a single logic regression model is desired, we may be better off using logic regression. The results of the Monte Carlo logic regression analysis suggest; however, that a model with predictors $X_1 = (\text{TP53(P72R)}_d \vee \text{CBS(I278T)}_r) \wedge \text{CD14}_d^c$, $X_2 = \text{TNFR1}_r \vee \text{APOC3(T3206G)}_d$, $X_3 = \text{MDM2}_d$, and $X_4 = \text{TNFR1}_d$ should fit the data reasonably well. The summary of this model can be found in Table V. This model includes the top five SNPs in Table II, the top four two-SNP interactions in Table III, and the top three three-SNP interactions in Table IV. In Table V, we summarize a logistic regression model with these four predictors for how many people in the data

TABLE V. Logistic regression model using predictors suggested by Monte Carlo logic regression

Predictor	Coefficient	SE	t-statistic	Frequency	Marginal odds ratio
1	−0.424				
$X_1 = (\text{TP53(P72R)}_d \vee \text{CBS(I278T)}_r) \wedge \text{CD14}^c_d$	0.804	0.150	5.38	48.5%	2.283
$X_2 = \text{TNFR1}_r \vee \text{APOC3(T3206G)}_d$	−1.817	0.479	−3.79	5.3%	0.153
X_3=MDM3_d	−0.545	0.231	−2.36	13.1%	0.599
X_4=TNFR1_d	−0.543	0.212	−2.57	15.8%	0.615

set this predictor was ''true,'' and the unadjusted odds ratio associated with each of these predictors. We note that some of these predictors are associated with substantially altered risk. The t-statistics do not adjust for multiple comparisons, and in fact, a conservative Bonferroni correction would leave each of these four predictors suggestive, but not statistically significant. Of additional interest is that the three-factor interaction in this model is considerably more significant than each of the three predictors by themselves: TP53(P72R)_d equals 1 for 53.1% of the people in this study and has an odds ratio of 1.499, CD14_d equals 1 for 21.7% of people in this study and has an odds ratio of 0.599, and CBS (I278T)_r equals 1 for 20.0% of people in this study and has an odds ratio of 1.450.

We postulate this model, as it seems a reasonable summary of the Monte Carlo logic regression analysis, and these SNPs and interactions appear worthwhile for studying in other populations using a more traditional hypothesis-testing approach. (An additional use of this model is for one of the simulation studies.) Which of these factors would stand up to such a second analysis remains open. In fact, we would be surprised if all predictors were confirmed but also if none were confirmed. As for almost all genes in the restenosis data, we have one or two SNPs, and in fact all of the SNPs involved in the model summarized in Table III come from different genes, and so the model in Table III does not relate to any particular haplotype. The inheritance pattern of this model is a mixture of dominant (SNPs with subscript $_d$) and recessive (SNPs with subscript $_r$).

COMPARISON TO STEPWISE SELECTION

Zee et al. [2002] used a stepwise logistic regression approach to select low-order interactions. Unfortunately, the way that they coded SNPs makes a direct comparison with our analysis impossible. In particular, the authors created numerical predictors by coding each SNP as 1, 2, or 3, corresponding to whether there were 0, 1, or 2 variant alleles, respectively, in the SNP. The authors selected a model which included both quadratic terms and interactions using this coding, without the inclusion of lower-order terms. This modeling strategy makes the selection of which SNPs are included in the model dependent on the coding. Nevertheless, from Table II we conclude that most of the SNPs selected in the current analysis agree with those selected by Zee et al. [2002].

Instead, to allow for a direct comparison between the proposed approach and stepwise logistic regression, we carried out a stepwise addition and deletion approach using the same 169 covariates as we used for the Monte Carlo logic regression. During the stepwise addition stage at each step, we selected among all covariates that were not yet in the model, and all interactions of two covariates that were in the model. After reaching a model with 40 functions, we proceeded with stepwise deletion. The best model size was selected using 10-fold likelihood cross-validation. This stepwise procedure is easily carried out using the Polyclass procedure [Kooperberg et al., 1997].

As can be seen from the model summary in Table VI, all but one of the covariates in this model were also among the most frequently selected SNPs and interactions by Monte Carlo logic regression shown in Tables II and III. However, there are some notable omissions, in particular the $\text{TNFR1}_r \vee \text{APOC3(T3206G)}_d$ interaction and TNFR1_d. Note that there are no three-way interactions in the stepwise model. This is no surprise, as there is little possibility to include an $A \times B \times C$ interaction in a logistic regression model that is selected using stepwise addition: a selection of such an interaction would require that first A, B, C, $A \times B$, $A \times C$, and $B \times C$ are in the model before we consider the three-way interaction. Thus, if the model in Table VI is the model from which stepwise addition is carried out, not a single three-way interaction would be considered.

TABLE VI. Logistic regression model selected using stepwise procedure

Predictor	Coefficient	SE	*t*-statistic
1	−0.565	0.188	−3.01
$IL4RA_r$	0.590	0.183	3.23
$TP53(P72R)_d$	0.432	0.152	2.84
$CD14_d$	−0.472	0.189	−2.49
$LPA(G121A)_d$	−1.454	0.652	−2.23
$APOC3(C1100T)_r$	−0.376	0.154	−2.45
$CBS(I278T)_r$	1.501	0.343	4.38
$GNB3_d$	−0.681	0.248	−2.75
$MDM2_d$	−0.398	0.284	−1.40
$IL4RA_r \times CBS(I278T)_r$	−1.271	0.407	−3.12
$CBS(I278T)_r \times MDM2_d$	−1.673	0.609	−2.75
$GNB3_d \times MDM2_d$	1.392	0.624	2.23

SIMULATION

Here we discuss two simulation studies. In the first simulation study, we further explore the performance of Monte Carlo logic regression on the restenosis data. In particular, we will compare the results of Monte Carlo logic regression on the actual restenosis data with the performance on data sets that were generated from a known model that fits the restenosis data. The second simulation study is more traditional, in that we generate data from several models, and compare the performance of Monte Carlo logic regression to two other methods.

RESTENOSIS DATA

Here we describe a simulation study to show that Monte Carlo logic regression usually identifies true interactions at a higher frequency than most noise interactions, assuming that signal is present in the data. Using the original SNP data, 25 sets of outcome data from the model summarized in Table V were generated. We use this model as our ''true'' model from which we generate new data. For each of these data, Monte Carlo logic regression with (K=3, a=1/$\sqrt{2}$) and (K=2, a=1/2) was carried out. The model from which the data were generated could be fit as a logic regression model with four logic trees, or as a logistic regression model with up to third-order interactions. In the existing simulation, we only consider modeling with up to two or up to three logic trees, so in fact the ''true'' model cannot be fit by logic regression. Monte Carlo logic regression can do a better job, as different logic trees will be in the model at different times during the iterations.

As in the analysis of the original data, for each data set and combination of (a, K), we ran three independent chains of length 5,000,000 after a burn-in of 10,000 iterations. In the simulations, we would expect the seven SNPs involved in the predictors listed in Table V, as well as $CBS(68bp\ ins)_r$ (as this SNP is virtually identical to $CBS(I278T)_r$) to occur considerably more frequently than other SNPs. Similarly, we would expect the six two-factor interactions between ($TP53(P72R)_d$ and $CBS(I278T)_r$), ($TP53(P72R)_d$ and $CD14_d$), ($CBS(I278T)_r$ and $CD14_d$), ($TP53(P72R)_d$ and $CBS(68bp\ ins)_r$), ($CBS(68bp\ ins)_r$ and $CD14_d$), and ($TNFR1_r$ and $APOC3(T3206G)_d$) and the two three-factor interactions between ($TP53(P72R)_d$, $CD14_d$, and $CBS(I278T)_r$) and ($TP53(P72R)_d$, $CD14_d$, and $CBS(68bp\ ins)_r$), to occur more often than other interactions. We thus counted how often these single SNPs were among the top selected SNPs, how often these two SNP interactions were among the top two SNP interactions, and how often these three SNP interactions were among the top three SNP interactions. As the results were virtually identical for (K=3, a=1/$\sqrt{2}$) and (K=2, a=1/2), we only show the results for (K=2, a=1/2). The results are summarized in Table VII. The results confirm our analysis. All the SNPs and interactions of SNPs that should have been selected were selected frequently. In fact, no other individual SNPs or interactions were selected that often. This suggests that if there is an effect in the data, the covariates involved with those effects are typically among the leading effects selected by Monte Carlo logic regression.

OTHER MODELS

To further compare the performance of Monte Carlo logic regression with stepwise logistic regression and logic regression, we generated data from five models. For each of the models, we generated 500 data sets with 1,000 individuals and 50 SNPs each. Each SNP was in Hardy-Weinberg equilibrium, with a probability of 0.25 of mutant alleles. We created linkage disequilibrium between SNPs 1–6 and between SNPs 7–12 by making correlations $cor(X_i, X_{i+1})$=0.95 for i=1, 2, 3, 4, 5, 7, 8, 9, 10, 11, and $cor(X_i, X_{i+1})$=0 otherwise. X_i and X_{i+2} are independent, given X_{i+1} for all i. Here X_i is the number of mutant alleles for the ith SNP. We recoded each SNP X_i in a dominant and a recessive binary predictor X_i^d and X_i^r, respectively,

as before. The five models which we generated for each data set were

$$\text{Mod 1:}\quad \text{logit}(P(Y=1)) = -2+1.2(X_3^d \wedge X_9^d),$$
$$\text{Mod 2:}\quad \text{logit}(P(Y=1)) = -1.4+0.4X_{13}^d,$$
$$\text{Mod 3:}\quad \text{logit}(P(Y=1)) = -1.4+0.6(X_{14}^d \vee X_{15}^d),$$
$$\text{Mod 4:}\quad \text{logit}(P(Y=1)) = -2+0.5X_{16}^d+0.25X_{17}^d +0.5(X_{16}^d \wedge X_{17}^d), \text{ and}$$
$$\text{Mod 5:}\quad \text{logit}(P(Y=1)) = -2+0.5X_{16}^d+1(X_{16}^d \wedge X_{17}^d).$$

The methods that we are comparing are logic regression with the number of trees and the number of leaves selected using cross-validation, stepwise logistic regression as implemented in Polyclass [Kooperberg et al., 1997], with the model complexity selected using cross-validation as well as with the model complexity selected using Bayesian Information Criterion (BIC), and Monte Carlo logic regression with $K=2$ and $a=2$, based on a single run of 5,000,000 iterations and 10,000 burn-ins. (We also ran computations for Monte Carlo logic regression with $K=3$ and $a=1/\sqrt{2}$, and as the results were very similar to those for $K=2$ and $a=1/2$, we omit those results.)

TABLE VII. Summary of simulation study for ($K=2$, $a=1/2$)[a]

Single SNPs			
$TP53(P72R)_d$	19	$CBS(I278T)_r$	14
$CD14_d$	19	$TNFR1_r$	13
$APOC3(T3206G)_d$	17	$CBS(68bp\ ins)_r$	10
$CBS(I278T)_r$	14	$MDM2_d$	9
Two-SNP interactions			
$TNFR1_r$		$APOC3(T3206G)_d$	14
$TP53(P72R)_d$		$CD14_d$	12
$TP53(P72R)_d$		$CBS(I278T)_r$	7
$CD14_d$		$CBS(I278T)_r$	6
$TP53(P72R)_d$		$CBS(68bp\ ins)_r$	6
$CD14_d$		$CBS(68bp\ ins)_r$	5
Three-SNP interactions			
$TP53(P72R)_d$	$CD14_d$	$CBS(68bp\ ins)_r$	9
$TP53(P72R)_d$	$CD14_d$	$CBS(I278T)_r$	9

[a]Tabulated are number of occurrences in top 8 (top 6, top 2) for single SNPs (two-way interactions, three-way interactions) that are part of correct model in 25 runs.

In Table VIII, we show how often out of 500 simulations each of the approaches selects the ''right'' interaction (main effect for model 2) with or without additional false positives. For Monte Carlo logic regression, we show how often the right interaction is selected if we require an interaction to be in at least 10% or 15% of the models after burn-in. In addition, for each model we determined the threshold for Monte Carlo logic regression to have the same number of simulations with the ''correct interaction with or without additional false'' as logic regression (LR). This approach is referred to as ''Monte Carlo LR (match-LR).'' The corresponding thresholds were 23.4%, 25.6%, 10.0%, 11.6%, and 22.0% for models 1, 2, 3, 4, and 5, respectively.

In Table VIII we also show the results for Polyclass with a penalty parameter selected separately for each model, such that the number of runs for which the ''correct interaction with or without additional false positives'' is selected exactly matches this number for regular logic regression. This choice is referred to as ''Polyclass (match-LR).'' The corresponding penalty parameters for Polyclass were 8.80, 5.22, 4.055, and 5.98 for models 2, 3, 4, and 5, respectively. For model 1, the smallest Polyclass penalty possible of 0 only yielded 103 models that included the correct interaction. For BIC, the Polyclass penalty is $\log(\text{sample size}) = \log(1{,}000) \approx 6.91$.

In Table IX, we show how often out of these 500 simulations only incorrect interactions or no interactions (main effects for model 2) were selected. Note that for each model/method

TABLE VIII. Frequency that correct interaction terms (main effects for model 2) are part of selected model in simulation study

	Models with or without false positives					Models without false positives				
Method	1	2	3	4	5	1	2	3	4	5
Logic regression	130	188	90	137	333	103	96	59	92	223
Polyclass (CV)	37	89	34	31	188	22	11	18	15	111
Polyclass (BIC)	72	259	43	43	294	69	120	40	40	274
Polyclass (match-LR)	N/A	188	90	137	333	N/A	151	57	26	268
Monte Carlo LR (10%)	240	286	90	147	372	22	126	68	114	203
Monte Carlo LR (15%)	190	245	80	129	352	56	149	66	109	238
Monte Carlo LR (match-LR)	130	188	90	137	333	81	157	68	112	269

combination, the numbers on the left side of Table VIII plus both sides of Table IX add up to 500. In Table X, we show how often logic regression and Monte Carlo logic regression selected an incorrect three-way interaction.

For model 1, there is linkage disequilibrium between the "correct" SNPs and nearby SNPs. As such, it is no surprise that all methods have a large number of false positives. In fact, the large majority of false positives for each method involve the SNPs with which SNPs 3 and 9 are correlated. When we fix the number of models that include the correct interaction with or without false positives, logic regression appears to outperform Monte Carlo logic regression. However, we believe that in a situation in which there is substantial linkage disequilibrium, it is advantageous if a model selection methodology identifies not just a single model, but several related models. A researcher would then be confronted with several comparable models that are closely related, and thereby would be drawn to the conclusion that the data at hand may sometimes not be sufficient to distinguish between several closely related SNPs. To investigate whether each of the methods would facilitate such an analysis, we show in Table XI for each of the methods how many of the 500 simulations for model 1 involve none, exactly one, and more than one interactions involving SNPs that either are the SNP that is associated with the outcome, or SNPs that are in linkage disequilibrium with those SNPs. We note from Table XI that both Polyclass and logic regression virtually never identify more than one of the interactions involving SNPs in linkage disequilibrium, while Monte Carlo logic regression does this frequently. As such, we believe that Monte Carlo logic regression is more useful in situations with a large amount of linkage disequilibrium than the other two approaches.

For model 2, there is only a fairly weak main effect. Not surprisingly, the methods with a high true-positive rate are also the methods with high false-positive rates. When we fix the number of models that include the correct interactions, we see that Polyclass and Monte Carlo logic regression perform similarly.

The true model 3 is a logic regression model with a fairly weak signal. Monte Carlo logic regression appears to slightly outperform regular logic regression and Polyclass.

Models 4 and 5 are traditional models that are more geared toward logistic regression, as these models include both main effects and interactions. The effect size for model 5 is stronger than for model 4. Surprisingly, for model 4, the Polyclass approach does much worse. The explanation is that with such a weak signal, the model selection will remove all predictors unless the penalty term is reduced so much that all sorts of noise predictors are included. For this model, Monte Carlo logic regression does better than all other approaches. In about 60–70% of the simulations where Monte Carlo logic regression with a threshold of 10%–15% selects a model, it selects exactly the right interaction. For model 5, Polyclass and Monte Carlo logic regression perform equivalently.

TABLE X. Frequency that incorrect three-way interactions are selected in simulation study

	Model				
Method	1	2	3	4	5
Logic regression	106	24	46	57	109
Monte Carlo LR (10%)	53	6	20	29	146
Monte Carlo LR (15%)	20	3	15	16	88
Monte Carlo LR (match-LR)	10	0	20	23	50

TABLE IX. Frequency that either only incorrect interactions or no interactions (main effects for model 2) are selected in simulation study

	Models with only incorrect interactions					Models with no interactions				
Method	1	2	3	4	5	1	2	3	4	5
Logic regression	332	66	72	76	33	38	246	338	287	134
Polyclass (CV)	101	36	39	48	19	362	375	427	421	293
Polyclass (BIC)	238	122	22	28	12	190	119	435	429	194
Polyclass (match-LR)	N/A	72	148	276	27	N/A	240	262	87	142
Monte Carlo LR (10%)	191	142	38	45	11	69	72	372	308	117
Monte Carlo LR (15%)	193	112	24	29	10	117	143	396	342	138
Monte Carlo LR (match-LR)	163	71	38	39	7	207	241	372	324	160

N/A: for Model 1 no polyclass penalty parameter could match the Logic Regression results.

TABLE XI. Frequency that interactions that are in linkage disequilibrium with correct interaction are selected for model 1

	Number of LD interactions		
Method	0	1	2
Logic regression	48	448	4
Polyclass (CV)	374	125	1
Polyclass (BIC)	207	292	1
Polyclass (match-LR)	N/A	N/A	N/A
Monte Carlo LR (10%)	78	71	351
Monte Carlo LR (15%)	123	146	231
Monte Carlo LR (match-LR)	213	213	74

N/A: for Model 1 no polyclass penalty parameter could match the Logic Regression results.

Table X shows how often each approach identified (incorrect) three-way interactions. For the stepwise logistic regression implementation, this is (virtually) impossible, since all three two-way interactions and all three main effects are required before a three-way interaction can be considered. Monte Carlo logic regression with a high threshold identifies very few high-order interactions, but with a lower threshold the number of three-way interactions, in particular for models 1 and 5, is quite a bit larger. For model 1 these interactions often involve several of the correlated SNPs, and may therefore in fact be desirable. For model 5, this is not true. In some sense, the real strong signal makes Monte Carlo logic regression (as well as logic regression) identify both the right interaction and some additional false positives.

In summary, in these simulations, Monte Carlo logic regression with a higher threshold performs very well, especially in situations where the signal is low. We believe that this is currently often the case in SNP studies, and thus that Monte Carlo logic regression may be a useful additional tool in analyzing such data. It is, unfortunately, not always clear what the right threshold is. In particular, we note that for model 3, the lower threshold worked slightly better than the higher threshold. In practice, the threshold will depend on both the sample size and the strength of the signal, and a simulation study like the one we did for the restenosis data may be a useful tool when interpreting the results.

CONCLUSIONS

Most recent methodological developments for the analysis of SNP association studies have been in reconstructing haplotypes for haplotype association analysis or on the selection of a limited number of tag-SNPs using haplotypes. There are situations, however, when the use of haplotypes may not necessarily be optimal, especially when tag-SNPs or a smaller number of SNPs on a larger number of genes are sequenced.

Logic regression was proposed to search directly for interactions between SNPs. As most data sets currently collected include several hundred to a thousand sequenced subjects, like any other analyses, logic regression might not have enough power to detect many interactions. This situation will likely improve for common diseases in the future as sequencing gets cheaper, and more subjects will be genotyped, but currently it prevents us from identifying subtle interactions. For rare diseases, data sets may never get sufficiently large. Thus it is important to identify all potential interactions on one data set, which hopefully can be validated by a follow-up study that will sequence fewer SNPs on a larger number of subjects. There is a need for methods that identify potential interactions on one data set that can be validated on others, sequencing fewer SNPs but more subjects. In this paper, we introduced Monte Carlo logic regression, an exploratory tool to generate lists of potentially important interactions. We believe that the combination of logic regression (the feature to directly search for interactions) with MCMC (the feature to create ensembles, not just a single model) will prove valuable.

ACKNOWLEDGMENTS

C.K. was supported in part by NIH grants CA 074841, CA 053996, and HL 74745. I.R. was supported in part by a Maryland Cigarette Restitution Fund Research Grant to the Johns Hopkins Medical Institutions and NIH grand CA 074841. The authors thank Li Hsu and Michael LeBlanc for many helpful discussions, and Jurg Ott, Klaus Lindpainter, and Robert Zee for generously sharing their data on post-PTCA restenosis.

ELECTRONIC DATABASE INFORMATION

Software for logic regression and Monte Carlo logic regression is available from http://bear.fhcrc.org/~ingor/logic.

REFERENCES

Breiman L, Friedman JH, Olshen RA, Stone CJ. 1984. Classification and regression trees. Belmont, CA: Wadsworth.

Denison DGT, Mallick BK, Smith AFM. 1998. Automatic Bayesian curve fitting. J R Stat Soc B 60:333–350.

Green PJ. 1995. Reversible jump Markov chain Monte Carlo computation and Bayesian model determination. Biometrika 82:711–732.

Hansen MH, Kooperberg C. 2002. Spline adaptation in extended linear models (with discussion). Stat Sci 17:2–51.

Hoh J, Ott J. 2003. Mathematical multi-locus approaches to localizing complex human trait genes. Nat Rev Genet 4:701–709.

Hoh J, Wille A, Ott J. 2001. Trimming, weighting, and grouping SNPs in human case-control association studies. Genome Res 11:2115–2119.

Johnson GCL, Esposito L, Barratt BJ, Smith AN, Heward J, Di Genova G, Ueda H, Cordell HJ, Eavens IE, Dudbridge F, Twells RCJ, Payne F, Hughes W, Nutland S, Stevens H, Carr P, Tuomilehto-Wolf E, Tuomilehto J, Gough SCL, Clayton DG, Todd JA. 2001. Haplotype tagging for the identification of common disease genes. Nat Genet 29:233–237.

Kooperberg C, Bose S, Stone CJ. 1997. Polychotomous regression. J Am Stat Assoc 92:117–127.

Kooperberg C, Ruczinski I, LeBlanc M, Hsu L. 2001. Sequence analysis using logic regression. Genet Epidemiol 21:626–631.

Nelson MR, Kardia SLR, Ferrell RE, Sing CF. 2001. A combinatorial partitioning method to identify multilocus genotype partitions that predict quantitative trait variation. Genome Res 11:458–470.

Ritchie MD, Hahn LW, Roodi N, Nailey R, Dupont WD, Parl FF, Moore JH. 2001. Multifactor-dimensionality reduction reveals high-order interactions among estrogen-metabolism genes in sporadic breast cancer. Am J Hum Genet 59:138–147.

Ruczinski I, Kooperberg C, LeBlanc M. 2003. Logic regression. J Comput Graph Stat 12:475–511.

Ruczinski I, Kooperberg C, LeBlanc M. 2004. Exploring interactions in high dimensional genomic data: an overview of logic regression, with applications. J Mult Anal 90: 178–195.

Sebastiani P, Lazarus R, Weiss ST, Kunkel LM, Kohane IS, Ramoni MF. 2003. Minimal haplotype tagging. Proc Natl Acad Sci USA 100:9900–9905.

Weiss SM, Clark AG. 2002. Linkage disequilibrium and the mapping of complex human traits. Trends Genet 18:19–24.

Zee RYL, Hoh J, Cheng S, Reynolds R, Grow MA, Silbergleit A, Walker K, Steiner L, Zangenberg G, Fernandez-Ortiz A, Macaya C, Pintor E, Fernandez-Cruz A, Ott J, Lindpaintner K. 2002. Multi-locus interactions predict risk for post-PTCA restenosis: an approach to the genetic analysis of common complex disease. Pharmacogenomics J 2:197–201.

APPENDIX

IMPLEMENTATION DETAILS OF MONTE CARLO LOGIC REGRESSION

We here provide some basic details about the implementation of Monte Carlo logic regression.

- Let K be the maximum number of logic trees in the model; let K_i be the number of nonzero logic trees at iteration i; thus $K_i \leq K$.
- Let L_{ji} be logic tree j at iteration i, $1 \leq j \leq K$, and let $L_{ji}=0$ if $j > K_i$. We code the shape of L_{ji} for all i and j, and count all nodes. Nonterminal nodes [terminology of Breiman et al., 1984] contain an operator, and terminal nodes contain a covariate or the complement of a covariate. The Monte Carlo logic regression model at iteration i (MCLR_i) is determined by K_i and L_{ji}, $1 \leq j \leq K_i$.
- Let s_{ji} be the number of terminal nodes in L_{ji}. If $L_{ji}=0$, we set $s_{ji}=0$. Let $S_i=\Sigma_j s_{ji}$ be the size of MCLR_i.
- Let ℓ_i be the likelihood of MCLR_i. This likelihood is obtained from regressing the response on the logic trees L_{ji}, $j=1,\ldots, K_i$ and the intercept.
- Let p_i be the prior of MCLR_i which is

$$p_i = (1-a)\times a^{S_i} \times \frac{1}{N(S_i)}$$

  where $N(S_i)$ is the number of possible MCLR models of size i. $N(S_i)$ depends on K, K_i, the maximum number of leaves in each tree, and the number of covariates. To compute $N(S_i)$, we precomputed the number of possible models of a particular size, ignoring the fact that in each terminal node, there can be c covariates and their complements, and we multiply this precomputed number by $(2c)^{S_i}$. There are some additional correction factors, as, for example, we reduce double counting by requiring that for every pair of adjacent terminal nodes, the leftmost node has a covariate with a lower index than the rightmost node.
- To propose a move, we first select an active logic tree, or we select an empty tree if $K_i < K$. We then determine which of the moves listed in Figure 1 are possible, and select one of those according to prespecified probabilities. (If all possible moves are possible, we take $P(\text{alternate leaf})=10/23$, $P(\text{alternate operator})=1/23$, and the probability of each of the other four-move types equal 4/23.) Depending on the move type, we may also need to select an operator and/or a predictor. While doing this, we keep track of the probability $q_{i \to i+1}$ that we would have in fact selected the move that we selected. This provides a candidate for MCLR_{i+1} which we refer to as MCLR'_{i+1}.
- We now compute p_{i+1} and ℓ_{i+1} corresponding to the proposed model, as well as the transition

probability $q_{i+1\to i}$ that we would make the reverse move if we started with MCLR'_{i+1}.

- Now set

$$r = \frac{p_{i+1}}{p_i} \times \frac{q_{i+1\to i}}{q_{i\to i+1}} \times \frac{\ell_{i+1}}{\ell_i}$$

and with probability $\min\{1, r\}$, we accept the proposed move and set $\text{MCLR}_{i+1} = \text{MCLR}'_{i+1}$. Otherwise, we set MCLR_{i+1}=MCLR_i.

- To make Monte Carlo logic regression a formal Bayesian procedure, we would need to specify a prior on the coefficients in the regression model, and we would generate a random coefficient vector at each step. Since our main interest is in which variables are in the models, rather than the models themselves, we forgo this step.