Available online at www.sciencedirect.com

SCIENCE DIRECT®

ELSEVIER

Journal of Multivariate Analysis 90 (2004) 178–195

Journal of Multivariate Analysis

http://www.elsevier.com/locate/jmva

Exploring interactions in high-dimensional genomic data: an overview of Logic Regression, with applications

Ingo Ruczinski,[a,*] Charles Kooperberg,[b] and Michael L. LeBlanc[b]

[a] Department of Biostatistics, Johns Hopkins University, 615 N. Wolfe St., Baltimore, MO 21205, USA
[b] Division of Public Health Sciences, Fred Hutchinson Cancer Research Center, 1100 Fairview Ave. N, Seattle, WA 98109, USA

Received 31 March 2003

Abstract

Logic Regression is an adaptive regression methodology mainly developed to explore high-order interactions in genomic data. Logic Regression is intended for situations where most of the covariates in the data to be analyzed are binary. The goal of Logic Regression is to find predictors that are Boolean (logical) combinations of the original predictors. In this article, we give an overview of the methodology and discuss some applications. We also describe the software for Logic Regression, which is available as an R and S-Plus package.
© 2004 Elsevier Inc. All rights reserved.

AMS 2000 subject classifications: 62J99

Keywords: Adaptive model selection; Boolean logic; Binary variables; Interactions; Single nucleotide polymorphisms

1. Introduction

Single base-pair differences, or single nucleotide polymorphisms (SNPs), are one form of natural sequence variation common to all genomes. SNPs are estimated to occur about every 1000 bases on average. SNPs in the coding region can lead to

*Corresponding author. Fax: +1-410-955-0958.
E-mail address: ingo@jhu.edu (I. Ruczinski).

0047-259X/$ - see front matter © 2004 Elsevier Inc. All rights reserved.
doi:10.1016/j.jmva.2004.02.010

amino acid substitutions and therefore impact the function of the encoded protein. Understanding how these SNPs relate to a disease outcome helps us understand the genetic contribution to that disease [5,7]. For many diseases the interactions between SNPs are thought to be particularly important [16].

Logic Regression is a generalized regression methodology intended for situations where most of the predictors are binary. Logic Regression searches for Boolean combinations of predictors in the entire space of such combinations, while being completely embedded in a regression framework, where the quality of the models is determined by the respective objective functions of the regression class. There is a wealth of approaches to building binary rules. Logic Regression stands apart from most methods in the computer science and machine learning literature in that it uses general Boolean expressions, a non-greedy search algorithm, and that it works in any regression framework.

As SNP data are effectively binary (see Section 4), Logic Regression is particularly useful for such data. In fact, Logic Regression has been developed with genomic applications in mind. In Section 2 we review the Logic Regression methodology, in Section 3 we compare Logic Regression to Classification and Regression Trees, and we show some applications to genomic data in Section 4. In Section 5 we describe the software that is available as a package for R and S-Plus. We conclude with a brief discussion in Section 6.

2. Methodology

Logic Regression [17,18] is intended for situations where most predictors are binary (0/1), and the goal is to find Boolean combinations of these predictors that are associated with an outcome variable. The main objective thereby is not to minimize the prediction error per se, but to explore models in a novel search space that might reveal important variables and interactions which would otherwise go unnoticed. First, assume that all predictors X_i are binary and write X_i instead of $\mathrm{Ind}(X_i = 1)$, and X_i^c instead of $\mathrm{Ind}(X_i = 0)$, where $\mathrm{Ind}(\cdot)$ is the usual indicator function. Any type of regression can be used, as all we need is a score (or cost or loss) function such as residual sum of squares in linear regression, log-likelihood in generalized regression, partial log-likelihood in Cox regression, or misclassification, that relates fitted values with the response. For example, assume that Y is a Bernoulli random variable, with $Y \in \{0, 1\}$. The simplest Logic Regression model is now $\hat{Y} = \mathrm{Ind}(L = 1)$, where L is any logic (Boolean) expression that involves the predictors X_i, such as $L = X_1$ or $L = X_1 \wedge (X_2^c \wedge (X_3 \vee X_4^c))$. Here, "$\wedge$" and "$\vee$" refer to the logic operations "and" and "or", respectively. Misclassification, i.e. $\sum(Y \neq \hat{Y})$, would be the score that we consider. If we want a regression equation of this form, the main problem is to find good candidates for L, as the collection of all possible logic terms is enormous.

It turns out to be very convenient to write logic expressions in tree form. For example, we can draw $X_1^c \wedge (X_2 \vee X_3)$ as the tree in the lower left panel of Fig. 1.

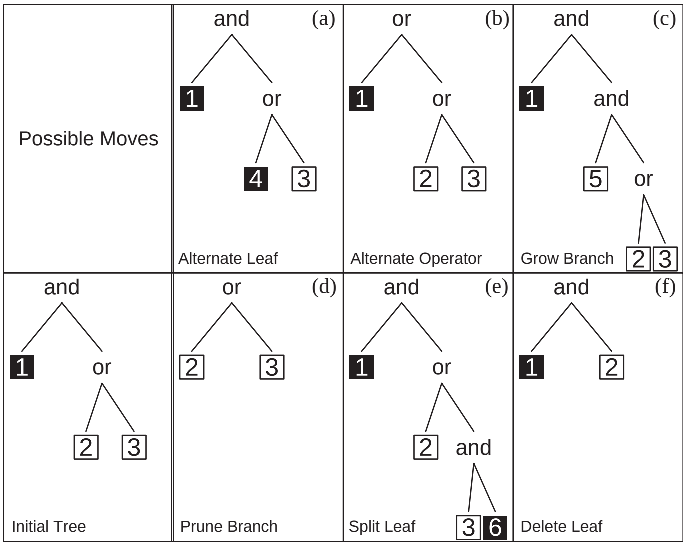


Fig. 1. Permissible moves for logic trees. The starting tree is in the panel on the lower left, the moves are illustrated in panels (a)–(f). A logic tree is evaluated by recursive substitution in a bottom-up fashion, as opposed to the top–down strategy in CART. The value (0 or 1) in the root node is the prediction of the tree for a specific case. We write i instead of X_i and we use white numbers on a black background to indicate the complement. For example, the initial tree in the lower left panel is X_1^c and (X_2 or X_3).

Using this "logic tree" representation, it is possible to obtain any other logic tree by a finite number of operations such as growing of branches, pruning of branches and changing of leaves (borrowing from the CART [4] terminology). In Fig. 1 we show the different types of moves that are currently implemented in the software. We should point out though that the rules obtained by the Logic Regression algorithm are distinctly different from those found by other tree based algorithms, such as CART [4]. We illustrate this point in more detail in Section 3.

For complicated problems, we may want to consider more than one logic tree at the same time. Thus, we can extend the classification model above (for example using a binomial likelihood) as

$$\operatorname{logit}(P(Y=1\,|\,\mathbf{X})) = \beta_0 + \sum_{j=1}^{m} \beta_j L_j, \tag{1}$$

where each of the L_j is a separate logic tree. Here the impact of multiple trees is linear on the logit scale.

Using the tree representation and the operations on logic trees as shown in Fig. 1, we can adaptively select L 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. This simulated annealing algorithm has similarities with the Bayesian CART algorithm [6], in which a CART tree is optimized stochastically. Both of these algorithms are distinct from the greedy algorithm employed by CART, in that at any stage they do not necessarily pick the move that improves the fit the most. Diagnostics, and a scheme that adjust the above-mentioned acceptance probabilities slowly enough during this algorithm, increases the chance that we will find (close to) the optimal model. An advantage of simulated annealing is that we are less likely to end up in a local extremum of the scoring function than if we used a non-stochastic algorithm. The price for this however is an increase in CPU time. Properties of the simulated annealing algorithm depend on Markov chain theory, and thus on the set of operations that can be applied to logic trees [1]. We should emphasize though that while the simulated annealing approach is sometimes computationally expensive, scalability is usually not a problem.

In practice not all predictors may be binary. Continuous predictors can still be included in Logic Regression models by allowing terms like $(X_i \leqslant a)$ to enter the model [17]. Alternatively, we can include continuous predictors in a regression model, in addition to logic terms, as we did for example for the GAW12 data (see Section 4).

Using model selection in addition to a stochastic model building strategy is of critical importance, as the logic model with the best score typically over-fits the data. For model selection we need a definition of model size. Usually, we define the size of a model as the total number of leaves in all logic trees combined, but other definitions are possible as well. The model selection consists of two steps. First, we find the best scoring models for various sizes. We can achieve this by prohibiting moves from a given model that would result in a model exceeding the pre-specified size, e.g. models of sizes up to eight leaves, allowing up to three trees. Among those, we pick the model that we consider to be "the best", either by comparing predictive performance or using permutation tests. For the predictive performance approach, if we have an abundance of data, we fit our models on one part of the data (training set), and validate them on the remainder (test set). For smaller data sets, we use cross-validation.

Permutation tests can generate test statistics for sequential hypothesis tests. First assume that a model of size s accurately models the data. We condition on this model, permuting all observations within the predicted classes that the logic trees define. We find the best Logic Regression model using these permuted data. We repeat this procedure, obtaining a histogram of scores that can be used to check the evidence against the null hypothesis that the model is indeed adequate. We compare this histogram to the best scoring model on the non-permuted data. If the scores of the non-permuted data are better than the scores on the permuted data the null hypothesis is rejected. Typically, we carry out these permutation tests for a sequence of Logic Regression models of different sizes. Usually, we observe that the locations of the histograms of the scores for different sizes (i.e. the mean scores) shift towards better scores until a tree size s_0 is reached, and then do not shift anymore. The best scoring model on the non-permuted data then looks like a draw from the distribution that the histograms approximate, and we select s_0 for our model. This procedure is described in more detail in [18], Section 4.2.

3. Comparison to classification and regression trees

As both CART and Logic Regression models are written in a "tree form", a natural question is how Logic Regression and CART compare. We investigated this in detail [17,18]. Here, we give a summary focusing on the form of the models, without actually discussing issues like model search, and illustrate this with an example below.

The CART algorithm generates a tree-based rule that can be written in disjunctive normal form (DNF [11]). For example,

$$\begin{aligned}&\text{if}\{[(X_1 \leqslant 1) \wedge (X_2 = 0) \wedge (X_3 > 6)] \vee \\ &\quad [(X_1 \leqslant 1) \wedge (X_2 = 0) \wedge (X_3 \leqslant 6) \wedge (X_4 = \text{``red''})]\} \\ &\quad \text{predict } \hat{Y} = 1 \end{aligned} \tag{2}$$

is in DNF, and could be part of a CART tree. In general, DNF rules are of the form $\bigvee_i E_i$, where $E_i = \bigwedge_j S_{ij}$, and the S_{ij} are simple relations, such as $(X_1 \leqslant 1)$. Note that not all rules in DNF can be directly written as a CART tree since, for example for the rules above, X_1 or its complement has to appear in every "and" statement in the DNF. The condition in (2) can be reduced to $\{(X_1 \leqslant 1) \wedge (X_2 = 0) \wedge [(X_3 > 6) \vee (X_4 = \text{``red''})]\}$ if Boolean expressions not in DNF are allowed. Clearly, it is a matter of taste whether general rules (summarized in bottom-up Logic Regression trees) or rules in DNF (summarized in top-down CART trees) are easier to interpret. It can be shown however that Boolean expressions (in DNF or not) and logic trees are equivalent in the sense that the classes of logic expressions they represent are the same. Using De Morgan's rules and standard Boolean operations, it can also be shown that a classification tree can be constructed from every Boolean expression, although these classification trees often result in very awkward looking constructs [17]. Hence, every classification tree can be written as a logic tree and vice versa. Knowing which rules each method prefers, it is straightforward to set up simulation studies in which either CART or Logic Regression performs "better". Instead we show a simple example of a data analysis in which both approaches provide useful, somewhat complementary, information. As it turns out, Logic Regression picks a model with two trees, another difference to CART, which always generates a single tree.

The data that we analyzed came from a study of aggressive histology non-Hodgkin's lymphoma that was carried out between 1985 and 1993, and was coordinated by the Southwest Oncology Group (SWOG). The study, which involved 899 untreated patients with advanced disease, was a four arm randomized clinical trial, comparing four different therapies. Of these 899 patients, 595 died during follow-up at the time of this analysis. The primary analysis [10] showed no difference between the treatments. For this disease, a widely used prognostic rule to define patient risk is based on "counting" the number of risk factors that a patient has, and to define the patient's risk status accordingly [19]. These risk factors, all of which have been identified in previous clinical studies as being associated with patient survival times, are whether (i) disease sites external to the lymph nodes are involved, (ii) the age of the patient is over 60, (iii) the disease stage is 3 or 4 (advanced stage), and (iv), whether the patient does have a high lactate dehydrogenase (LDH) level (associated with tumor bulk). While certainly other covariates are associated with patient survival, in the current analysis we focus only on these four binary risk factors as predictors for survival. The goal of this analysis is thus to investigate how rules identified by Logic Regression and survival trees compare to the IPI index [19]. In our analysis, we use Logic Regression with the Cox partial likelihood [8] as the scoring function, and the survival trees of LeBlanc and Crowley [15] as the regression tree approach.

Using the model selection tools described in Section 2, we applied Logic Regression using partial likelihood as the score for a proportional hazards model. The regression function that was selected was $g(\mathbf{X}) = \exp\{0.49L_1 + 0.44L_2\}$, where $L_1 = [((\text{not stage } 3/4) \wedge (\text{extra nodes})) \vee (\text{age} > 60)]$, and $L_2 = [(\text{high LDH}) \wedge ((\text{stage } 3/4) \vee (\text{age} \leqslant 60))]$, as shown in Fig. 2. The two factors L_1 and L_2 can be interpreted as alternative risk factors. We refer to $\log(g(\mathbf{X}))$ as the log-relative risk. As the coefficients 0.49 of L_1 and 0.44 of L_2 are quite similar in this case (both have standard errors of about 0.08), we can basically add the number of Logic Regression risk factors (here, the trees), similar to what is traditionally done with individual prognostic factors.

The pruned survival tree for these data is shown in Fig. 3. The numbers under the node indicate the number of patients in that node, and the fitted log-relative risk of

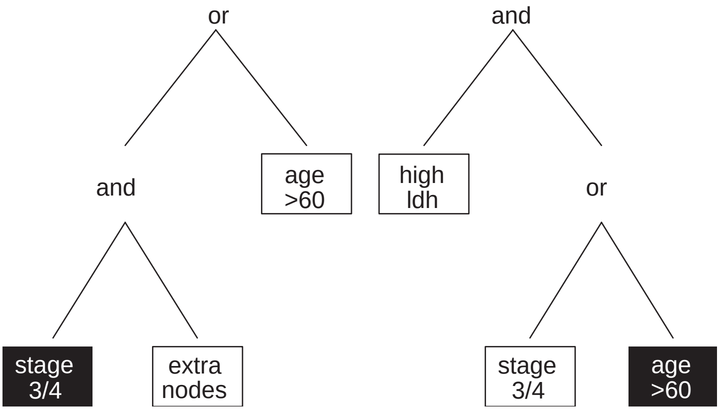


Fig. 2. Fitted logic trees for the lymphoma data. Variables that are printed white on a black background are the complement of those indicated. Patients for whom the left tree is true have an increase in their fitted log-relative risk of 0.49; patients for whom the right tree is true have an increase in their fitted log-relative risk of 0.44.

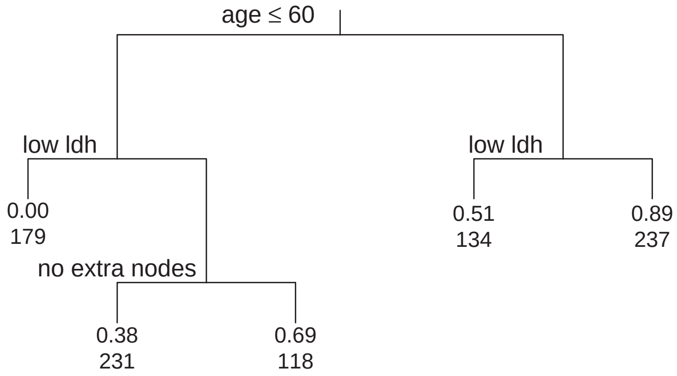


Fig. 3. Fitted survival tree for the lymphoma data. The first number at each node is the log-relative risk relative to the left most node with low LDH and age $\leqslant 60$, and the second number is the number of patients in that node.

those patients relative to the left most node of patients with age under 60 and no high LDH. The pruned survival tree can be written in a proportional hazards model with regression function $g(\mathbf{X}) = \exp\{0.38N_2 + 0.69N_3 + 0.51N_4 + 0.89N_5\}$, where N_j is an indicator function for a patient being in the jth node (counting from left to right) in the survival tree of Fig. 3.

For the IPI index we fitted a proportional hazards model using indicators for the number of risk factors that a particular patient has. This yielded the model $g(\mathbf{X}) = \exp\{0.47R_1 + 0.69R_2 + 1.03R_3 + 1.40R_4\}$, where R_j is an indicator function for a patient having exactly j risk factors.

As the four risk factors together define just 16 strata, it is possible to compare the three approaches (counting risk factors, Logic Regression, survival trees) to a proportional hazards model fitting a parameter to each stratum separately. In Table 1 we show the number of patients in each stratum as well as the number of risk factors, and the fitted log-relative risk for each of the modeling approaches.

Fig. 4 visualizes this table. The upper left panel shows the fitted log-relative risks for the 16 cells using separate strata. The other three panels show how counting risk factors, Logic Regression, and survival trees "smooth" over those 16 cells in their respective model spaces. Counting risk factors seems to have a somewhat large discrepancy with the separate strata approach in the bin where only stage 3/4 is false, however this cell contained only 2 observations. While the trees from the logic model may not be that intuitive at first, it is noteworthy that they predict the highest relative risk for the three bins that were also indicated by fitting separate strata.

Table 2 provides some summary statistics for the three modeling approaches that combine strata, and for the model with 16 different strata. As can be seen from this table, the model with 16 strata has, naturally, the largest log-likelihood. The other three approaches yield a very similar score. The picture changes if we take into account how many parameters are involved in each model. The Akaike Information Criterion (AIC) [2] for the model with separate strata is much larger than the other

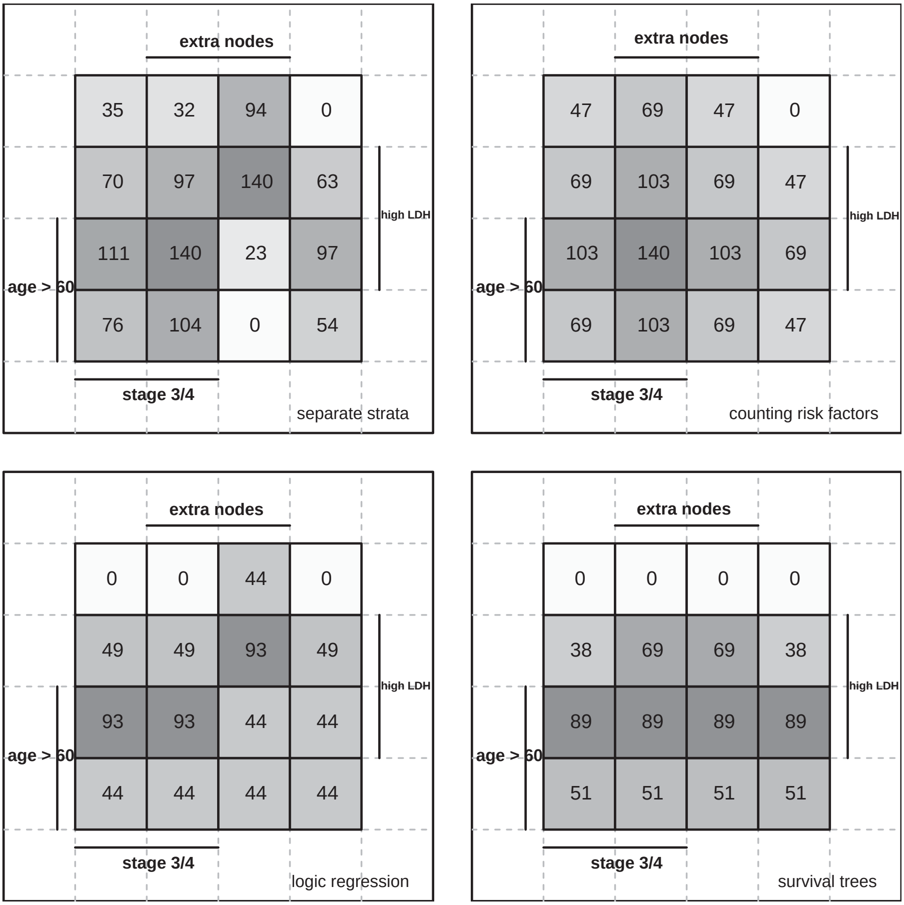
Table 1
Comparison of various approaches to identifying risk factors for Lymphoma

— Risk factors —						— Log-relative risk —			
Extra Nodes	Age >60	Stage 3 or 4	High LDH	*n*	% Died	Separate strata	Counting risk factors	Logic regression	Survival trees
0 risk factors									
.	.	.	.	29	38	0.00	0.00	0.00	0.00
1 risk factor									
.	.	.	Y	53	58	0.63	0.47	0.49	0.38
.	.	Y	.	94	51	0.35	0.47	0.00	0.00
.	Y	.	.	14	57	0.54	0.47	0.44	0.51
Y	.	.	.	6	67	0.94	0.47	0.44	0.00
2 risk factors									
.	.	Y	Y	178	62	0.70	0.69	0.49	0.38
.	Y	.	Y	28	68	0.97	0.69	0.44	0.89
.	Y	Y	.	71	72	0.76	0.69	0.44	0.51
Y	.	.	Y	7	86	1.40	0.69	0.93	0.69
Y	.	Y	.	50	50	0.32	0.69	0.00	0.00
Y	Y	.	.	3	33	−0.14	0.69	0.44	0.51
3 risk factors									
.	Y	Y	Y	116	78	1.11	1.03	0.93	0.89
Y	.	Y	Y	111	68	0.97	1.03	0.49	0.69
Y	Y	.	Y	2	50	0.23	1.03	0.44	0.89
Y	Y	Y	.	46	78	1.04	1.03	0.44	0.51
4 risk factors									
Y	Y	Y	Y	91	86	1.40	1.40	0.93	0.89

The "separate strata" model fits a parameter in the proportional hazards model to each of the $2^4 = 16$ different combinations of the risk factors; the "counting risk factors" model combines the strata with the same number of risk factors, fitting separate parameters for the people with 0, 1, 2, 3, and 4 risk factors; the "logic regression" and "survival trees" models are described in the text. Log-relative risks are relative to the group with no risk factors; the choice of the reference group is arbitrary.

three models, the counting risk factors approach and the survival trees have a similar AIC, and the Logic Regression approach has the lowest (best) AIC. As the IPI was not established on this data set, it is somewhat surprising that it still performs as well as the survival tree approach, which, like Logic Regression, determines the model adaptively.

Logic Regression presents a simple rule, defining three strata with the lowest risk, 10 intermediate strata, and three strata with high risk. Survival trees, are a bit more subtle in this situation, and separate the five strata with the worst survival into two different groups. We cannot say whether survival trees or Logic Regression is "better" in some sense: the example simply served as an illustration what kind of information can be extracted from the data with the different methods. In Section 4.1 we show an example where Logic Regression was able to pick out some signal where other methods failed to do so.



separate strata

35	32	94	0
70	97	140	63
111	140	23	97
76	104	0	54

counting risk factors

47	69	47	0
69	103	69	47
103	140	103	69
69	103	69	47

logic regression

0	0	44	0
49	49	93	49
93	93	44	44
44	44	44	44

survival trees

0	0	0	0
38	69	69	38
89	89	89	89
51	51	51	51

Fig. 4. The log-relative risks fitted by separate strata, counting risk factors, logic regression, and survival trees. The four binary predictors generate 16 cells for all true/false combinations. A predictor being true is indicated by a bar, the absence of a bar indicates that the predictor is false. For example, the lower two rows in each panel are the 8 cells for which the subjects were older than 60 years, and the upper two rows in each panel are the 8 cells for which the subjects were 60 years of age or less. The upper right corner is the cell for which all binary predictors were false. The grey scales reflect the fitted log-relative risks, and were chosen such that the minima and maxima, respectively, have the same color in each panel.

4. Genomic applications

While Logic Regression has been used for non-genomic applications [9], many of the developments of Logic Regression were motivated by statistical issues in analyzing genomic data. An initial application of Logic Regression was to the simulated SNP data from the twelfth genetic analysis workshop [20]. This application has been discussed in detail elsewhere [14,18], so we only provide a brief summary here. We also discuss an application to SNP data on post PTCA restenosis [22].

Table 2
Summary statistics for the various approaches to identifying risk factors for Lymphoma

Model	Partial likelihood	Parameters	AIC
Separate strata	−3721.60	15	7473.20
Counting risk factors	−3728.02	4	7464.04
Logic regression	−3728.49	2	7460.99
Survival trees	−3728.32	4	7464.62

See Table 1 and the text for a detailed description of the models.

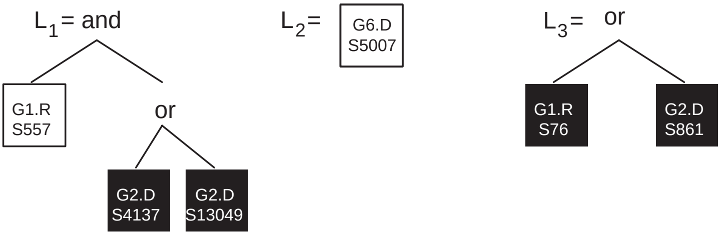


Fig. 5. Fitted Logic Regression model with three trees and six leaves for the affected state in the GAW data. Variables that are printed white on a black background are the complement of those indicated.

4.1. Results from the 12th genetic analysis workshop (GAW 12)

Each human cell contains a pair of every autosome, and typically only two of the four possible nucleotides occur at one particular SNP. Thus we can think of a SNP as a variable X taking values 0, 1 and 2 (for example, corresponding to the nucleotide pairs AA, AT/TA and TT, respectively). We can re-code this variable corresponding to a dominant gene as $X_d = 1$ if $X \geqslant 1$ and $X_d = 0$ if $X = 0$, and as a recessive gene as $X_r = 1$ if $X = 2$ and $X_r = 0$ if $X \leqslant 1$. This way, we generate $2p$ binary predictors out of p SNPs. The Logic Regression algorithm is now well suited to relate these binary predictors with a disease outcome.

For the GAW12 workshop the data were simulated under the model of a common disease. Prevalence increases with age, and the disease was more common in females than in males. A total of 50 replicate datasets were generated, each of which consisted of 23 extended pedigrees with 1497 individuals (1000 living). Each living subject had data on affection status, age at last exam, age at onset if affected, marker genotype data for a 1cM autosomal genome screen, and sequence data on six candidate genes, as well as five quantitative traits and two environmental factors. We applied Logic Regression to the sequence data [14]. In our analysis we used one copy of the data set as training data, another copy as test data, and ignored the remaining 48 replicates. In the sequence data, there were a combined total of 694 SNPs on six genes with at least 2% mutations.

As response we used the affected status and the deviance as scoring function, ignoring pedigree structure. The model selection yielded a model with three trees and six leaves, shown in Fig. 5. Here *Gi.D.Sj* refers to site j on gene i, using dominant coding, i.e. $Gi.D.Sj = 1$ if at least one variant allele exists. Similarly, $Gi.R.Sj$ refers to site j on gene i, using recessive coding. The logistic regression model corresponding to this Logic Regression model was

$$\begin{aligned}\text{logit}(\text{P}(\text{affected})) = {} & 0.44 + 0.005 \times \text{env}_1 - 0.27 \times \text{env}_2 + 1.98 \times \text{gender} \\ & - 2.09 \times L_1 + 1.00 \times L_2 - 2.82 \times L_3, \end{aligned} \tag{3}$$

where env_1 and env_2 denote the environmental factors.

Since the GAW data were simulated, we knew the correct solution. As it turned out, the Logic Regression algorithm picked exactly those mutational sites on genes 1 and 6 that were used in generating the data, and a number of sites on gene 2 where there were multiple mutational hits. Various methods were used by other groups to analyze the data, including (generalized) linear models, mixed models, annealing based best subset regression, and multivariate adaptive regression splines (MARS [12]). While the results of all the groups are not fully comparable (see [21] for a full discussion), the Logic Regression approach stood out as it detected the correct interaction between genes 1 and 2 and did not include any spurious sites.

Subsequently, we have also analyzed the data with CART [4] and Random Forests [3]. Similar to the Logic Regression approach, we grew a CART tree on the training set (1000 cases), and evaluated the sequence of the CART subtrees on the test set (also 1000 cases). The best tree had 8 splits in a total of four variables. The variables in this subtree were gender, $G1.R.S557$, $G1.R.S2923$, and env_1. In the Logic Regression model the separately fitted variable env_2 was not statistically significant, so it is no surprise that it did not appear in the CART tree. And while $G1.R.S2923$ is a false positive, it is 98% identical to $G1.D.S557$. However, none of the sites on gene 2 and gene 6 were chosen. While Random Forests do not carry out a model selection, they do provide measures of variable importance that can be used as guidance in variable selection. Since the Random Forest methodology does not include a model selection process equivalent to CART or Logic Regression, we ran the software on the training data alone, and also on the training data and test data combined. Considering the two measures of variable importance implemented in R (see the randomForest manual), clearly outstanding according to both measures were the variables gender, $G1.R.S557$, and env_1. The variables $G1.D.S557$ and $G1.R.S596$ (a false positive, but 89.9% identical to $G1.R.S557$) were also considered important, although less so then the three variables mentioned above. Again, none of the sites on either gene 2 or gene 6 were considered important. Since the number of predictor variables was large, we carried out three independent runs of the Random Forest algorithm growing 10,000, 25,000, and 100,000 trees. The results were virtually identical, indicating that we used sufficiently many iterations.

CART and Logic Regression had equivalent misclassification rates for the test set, 192 and 194 out of the 1000 cases, respectively. Interestingly, the misclassification rate on the test data for the Random Forest procedure was 216 out of 1000 cases, somewhat higher than the other approaches. The error rate using the out-of-bag portion of the training data was 20.2%, and 20.4% on the training and test data combined. These results were rather surprising since we expected a lower misclassification rate for Random Forests. We used the currently available R package (version 3.9-6) which might not be as flexible as the original code. Although we tried to set the parameters in the function randomForest() to allow large trees, the size of the trees in the forest might not have been sufficient. This suspicion is somewhat supported by the fact that we had to re-set the default tree control parameters in CART as well. The CART tree with seven splits misclassified 215 out of 1000 cases on the test data, compared to 192 cases for the selected tree with 8 splits.

4.2. Post PTCA restenosis

Here we discuss our analysis of a SNP study on the incidence of restenosis following percutaneous transluminal coronary angioplasty (PTCA). These data were reported and analyzed by Zee et al. [22], and were generously provided to us by Dr. Jurg Ott. Between March 1995 and March 1997, 779 subjects were enrolled in this study. The patients all underwent a successful PTCA. Of these 779 patients, 342 ("cases") developed restenosis within six months, the other 437 ("controls") did not. For the 779 patients, 94 SNPs in 62 candidate genes were sequenced. Of these, 6 SNPs had no variation in the sample. Among the remaining 88 SNPs, 7 had one variant allele, and 81 had two variant alleles. We recoded each SNP with two variant alleles by two binary predictors, similar to what we did for the GAW data, yielding a total of 169 binary predictors. We fitted Logic Regression models with up to three logic trees and up to eight leaves in all trees combined. Model selection was carried out using cross-validation and permutation tests.

Unfortunately the way that the SNPs are coded in [22] makes a direct comparison with our analysis impossible. In particular, the authors chose 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. Further, the authors selected a model which included both quadratic and interactions terms using the above coding, without the inclusion of lower order terms. Because of these modeling choices we have to restrict our comparison to which SNPs are (or are not) in the respective models.

Using the binary variables generated from the SNPs as predictors, an initial exploratory data analysis did not show any association between the predictors and the response (cases/controls). We calculated the t-statistics for each of the 169 binary predictors using simple logistic regression. The largest absolute t-statistic was 2.82 (corresponding to a p-value of 0.005, which is not significant after a Bonferroni correction). In addition, the cross-validated CART trees contained no split. However, we can detect some association between SNPs and disease status when we examine the results from the Logic Regression analysis.

We fitted Logic Regression models with up to three logic trees and up to eight leaves in all trees combined. Fig. 6 shows the results from the permutation tests using a single logic tree. The histograms of the permutation scores, used as reference distributions, keep shifting until we condition on a tree of size four, indicating the best model size. While we can conclude that there is an association between the predictors and the response, it is noteworthy that 66 out of 500 permutation scores

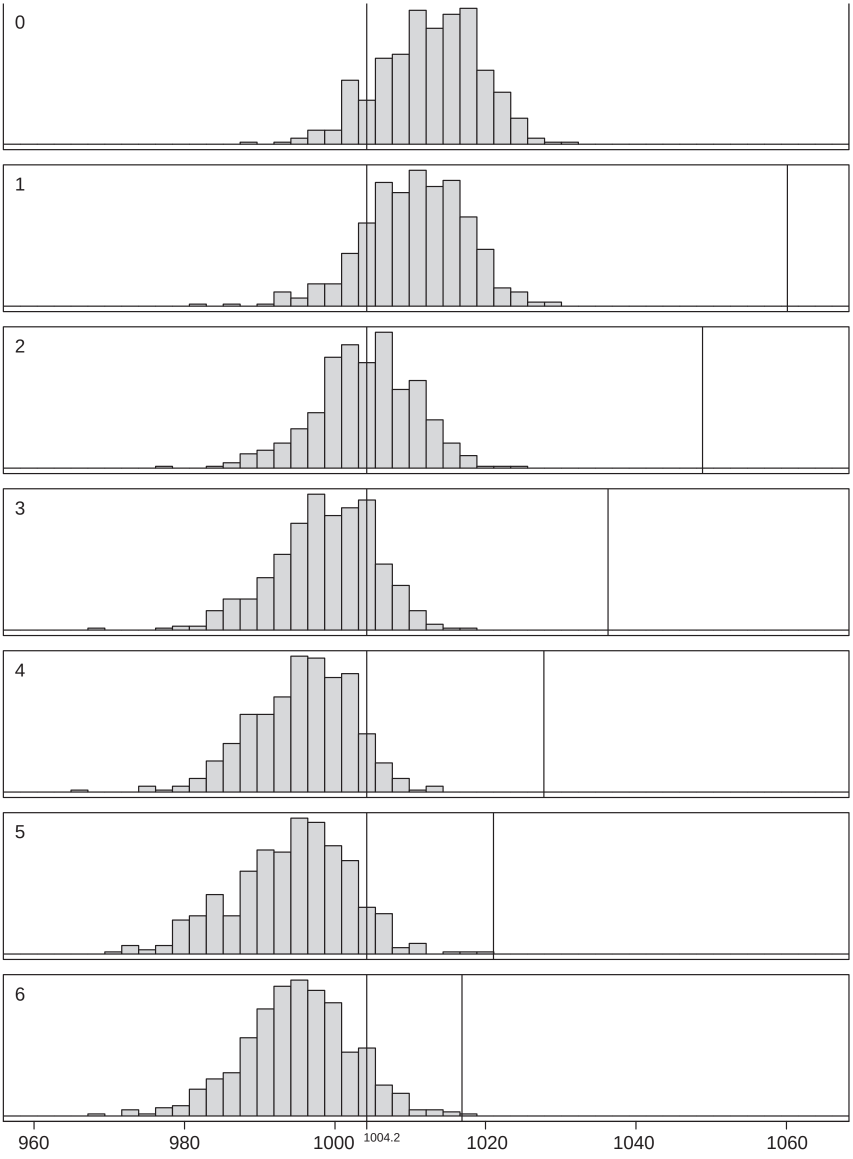


Fig. 6. The deviances obtained in the permutation tests for model selection for the post PTCA restenosis data, using a single logic tree, and 500 permutations per model size. Permutation scores were obtained conditioning on 0,1,2,3,4,5 and 6 leaves, indicated by the number in the upper left corner of each panel. The solid bars at deviance 1004.2 indicate the best overall score found using the original (non-permuted) data. The bars in the panels moving to the left show the best score found for models with one tree and the respective size, using the original (non-permuted) data. Since the mean of the permutation scores improves until we condition on the model of size 4, we pick a logic tree with four leaves.

(13.2%) in the null model test exceed the best scoring model (Fig. 6, top panel), shedding some light upon the weakness of the signal.

In Fig. 7 we show the results of the Logic Regression cross-validation. These results are somewhat inconclusive as to which model is best. In particular, the difference in predictive performance (as measured by the cross-validation score) between the null-model, models with one logic tree with up to five leaves, models with two logic trees with up to four leaves, and the model with three logic trees and three leaves, are statistically insignificant after averaging scores over several repeated cross-validations. In addition, the differences in deviance between some of the larger models are still substantial, suggesting that while these larger models may not *predict* as well as the smaller models, the combinations in these models are still associated with post PTCA restenosis. As both for the training score and the cross-validation score the models with fewer trees tend to perform better than the models with more trees and the same number of leaves, we conclude that an additive model may not be the most appropriate model for these data. Also note that the model of size zero (just fitting an intercept) has the lowest estimated predictive error, which is in agreement with the results from the CART analysis.

In Fig. 8 we show the fitted Logic Regression model with one tree and four leaves. Some further analysis suggested that the dominantly coded SNPs on genes TP53 and CD14 are the strongest predictors, while some additional SNPs have associations that are almost as strong as the recessively coded SNPs on genes ADRB3 and CBS, suggesting why the model selection did not select one definitive model. This is in

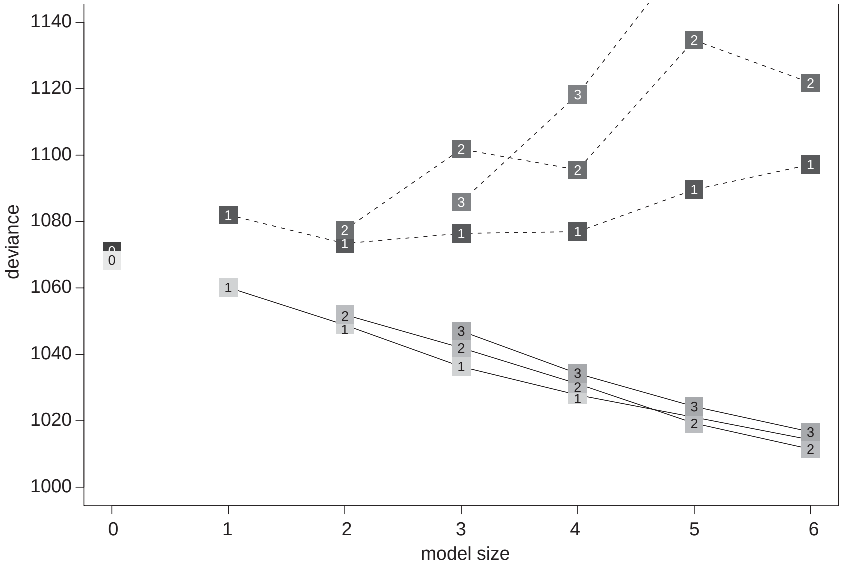


Fig. 7. Cross-validation results for the post PTCA restenosis data. The numbers in the squares indicate the number of logic trees associated with the model; the solid lines are for the training data, the dashed lines are for the test data. Both training and test data score averages from 10-fold cross-validation are rescaled to the complete sample size (i.e. multiplied by $\frac{10}{9}$ and 10, respectively).

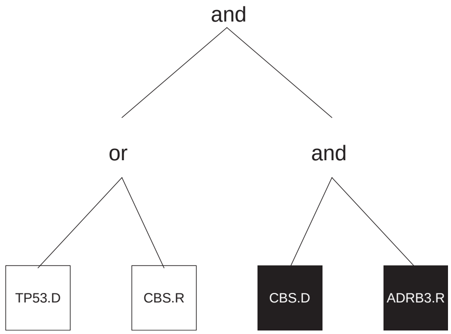


Fig. 8. Fitted logic tree for the post PTCA restenosis data. Variables that are printed white on a black background are the complement of those indicated; a D refers to dominant coding and an R to recessive coding of the particular SNP.

partial agreement with the results in the analysis of [22], who identified a significant association of seven SNPs with the disease outcome. In their model, gene CBS is included as a main effect, and genes TP53 and CD14 appear in quadratic effects. CBS and CD14 also appear in higher order interaction terms.

In summary, Logic Regression finds some suggestions of associations between SNPs and post PTCA restenosis, though the predictive value of these SNPs is low. This is not totally unexpected, as most SNPs are not expected to be associated with large risks, and in this particular data set the number of cases and controls is quite low. (The authors of this paper are involved in several ongoing studies that eventually will involve a few thousand records.) Nevertheless, the suggestions of association are of importance, as those can be tested on data from other sources.

5. Software

The software used for the analyses in this manuscript is freely available from the Logic Regression website at http://bear.fhcrc.org/~ingor/logic, and can be downloaded as a package for R or S-Plus. The core of the Logic Regression code is in Fortran 77, which is called by R or S-Plus. Current options include fitting one (large) Logic Regression model, fitting Logic Regression models of pre-specified sizes, carrying out cross-validation, and various permutation tests for model selection. Plotting functions to display the results are included. All functions have extensive help files. Currently the Logic Regression methodology has scoring functions for linear regression (residual sum of squares), logistic regression (deviance), classification (misclassification), and proportional hazards models (partial likelihood). A feature of the Logic Regression methodology is that it is possible to extend the method to write ones own scoring function if this is necessary. We also implemented a greedy stepwise algorithm, as an alternative to simulated annealing, and plan to release it with a future version of the software.

6. Discussion

Within the last 5 years, the amount of SNP and other genomic data generated has increased dramatically. We developed Logic Regression as a tool to analyze some of those data, in particular when most of the predictors are binary and interactions are of primary concern. While Logic Regression models will often give good predictions, one of the main strengths of Logic Regression is the ease of interpretation of the selected models. This sets it apart from methods like neural networks, where the emphasis is almost completely on prediction rather than interpretation. Many other methods for high-dimensional function estimation fall somewhere in between these two extremes.

We currently work on enhancements of the methodology and the software. One of those extensions we investigate is how to generate ensembles of good scoring models. After exploring a sequence of candidate models by fitting models of various sizes, we pick the model size using cross-validation or permutation tests. This yields a single model. However, often there are many models that adequately explain the signal in the data. In particular, SNPs that are close in the DNA sequence tend to be highly correlated. One approach to address this issue is to formulate our prior beliefs about the models and employ the Bayesian machinery to create a posterior distribution over features of the model space. Another approach takes advantage of the fact that if we carry out simulated annealing at a fixed temperature, we run a Metropolis–Hastings algorithm and generate (dependent) samples from its limiting distribution. Therefore, ensembles of good scoring models can be generated if the temperature is chosen adequately.

There is another advantage to the above described extensions. Classification trees have been known to be rather unstable, and used in its original form do not have high predictive power [13]. Bagging and (tree-based) boosting are extensions of classification trees that usually generate much better predictors. The price to pay for this gain of predictive power is the loss in interpretability. Presumably boosting and bagging could also improve the predictive performance of Logic Regression in a similar way. Logic models are also unstable, but we emphasize the interpretability of logic models as one of its most desirable and important properties, and are not concerned with similar resampling or reweighting schemes. However, one useful feature of bagging is that it generates a measure of variable importance. The above-mentioned extensions of Logic Regression also include this feature. Since, we explore all possible Boolean combinations, we can also generate a list of important interactions.

Another concern is that most SNP data contain a fair number of missing values. At the moment, we use imputation methods before applying our methodology: given the observed data, we model a distribution for the missing values, and simulate from this distribution to fill in the missing values. Besides searching for improvements in this strategy, we also explore alternatives that do not depend on such up-front imputations, for example methods similar to the CART surrogate split strategy.

Acknowledgments

Ingo Ruczinski was supported in part by NIH Grant CA 83079 and through the support of the Maryland Cigarette Restitution Fund Research Grant to the Johns Hopkins Medical Institutions. Charles Kooperberg was supported in part by NIH Grants CA 74841 and CA 53996. Michael LeBlanc was supported in part by NIH Grants CA 90998 and CA 53996. The GAW workshop is supported by NIH Grant GM 31375. The authors thank Drs. Klaus Lindpainter, Jurg Ott, and Robert Zee for sharing the data on post-PTCA restenosis, and the organizers of the Genetic Analysis Workshop for their permission to use the GAW12 data in this paper.

References

[1] E.H.L. Aarts, J.H.M. Korst, Simulated Annealing and Boltzmann Machines, Wiley, New York, 1989.

[2] H. Akaike, A new look at the statistical model identification, IEEE Trans. Ac. 19 (1974) 716–723.

[3] L. Breiman, Random forests, Machine Learning 45 (2001) 5–32.

[4] L. Breiman, J.H. Friedman, R.A. Olshen, C.J. Stone, Classification and Regression Trees, Wadsworth, Belmont, CA, 1984.

[5] A. Chakravarti, Population genetics—making sense out of sequence, Nat. Gen. 21 (1999) S56–S60.

[6] H. Chipman, E. George, R. McCulloch, Bayesian cart model search (with discussion), J. Amer. Statist. Assoc. 93 (1998) 935–960.

[7] F.S. Collins, A. Patrinos, E. Jordan, A. Chakravarti, L.W.R. Gesteland, New goals for the US human genome project: 1998–2003, Science 282 (1998) 683–689.

[8] D.R. Cox, Regression models and life tables (with discussion), J. Roy. Statist. Soc. B 34 (1972) 187–220.

[9] R. Etzioni, C. Kooperberg, M. Pepe, R. Smith, P.H. Gann, Combining biomarkers to detect disease with application to prostate cancer, Biostatist. 4 (2003) 523–538.

[10] R.I. Fisher, E.R. Gaynor, S. Dahlberg, M.M. Oken, E.M.M.T.M. Grogan, J.H. Glick, C.A. Coltman, T.P. Miller, Comparison of a standard regimen (chop) with three intensive chemotherapy regimens for advanced non-hodgkin's lymphoma, N. Engl. J. Med. 328 (1993) 1002–1006.

[11] H. Fleischer, M. Tavel, J. Yeager, Exclusive-or representation of boolean functions, IBM J. Res. Dev. 25 (1983) 412–416.

[12] J.H. Friedman, Multivariate adaptive regression splines (with discussion), Ann. Statist. 19 (1991) 1–141.

[13] T. Hastie, R. Tibshirani, J. Friedman, The Elements of Statistical Learning, Springer, New York, 2001.

[14] C. Kooperberg, I. Ruczinski, M. LeBlanc, L. Hsu, Sequence analysis using logic regression, Gen. Epi 21 (2001) S626–S631.

[15] M. LeBlanc, J. Crowley, Survival trees by goodness of split, J. Amer. Statist. Assoc. 88 (1993) 457–467.

[16] P.R. Lucek, J. Ott, Neural network analysis of complex traits, Gen. Epi. 14 (1997) S1101–S1106.

[17] I. Ruczinski, Logic regression and statistical issues related to the protein folding problem, Ph.D. Thesis, University of Washington, Seattle, WA 98195, 2000.

[18] I. Ruczinski, C. Kooperberg, M. LeBlanc, Logic regression, J. Comput. Graph. Statist. 12 (2003) 475–511.

[19] The International Non-Hodgkin's Lymphoma Prognostic Factors Project, A predictive model for aggressive non-hodgkin's lymphoma, N. Engl. J. Med. 329 (1993) 987–994.

[20] E.M. Wijsman, L. Almasy, C.I. Amos, I. Borecki, C.T. Falk, T.M. King, M.M. Martinez, D. Meyers, R. Neuman, J.M. Olson, S. Rich, M.A. Spence, D.C. Thomas, V.J. Vieland, J.S. Witte,

J.W. MacCluer (Eds.), Analysis of complex genetic traits: Applications to asthma and simulated data, Gen. Epi. 21 (2001) (S1).

[21] J.S. Witte, B.A. Fijal, Introduction: analysis of sequence data and population structure, Gen. Epi. 21 (2001) S600–S601.

[22] R.Y.L. Zee, J. Hoh, S. Cheng, R. Reynolds, A.S.M A Grow, K. Walker, L. Steiner, A.F.-O.G Zangenberg, C. Macaya, E. Pintor, A. Fernandez-Cruz, K.L.J Ott, Multi-locus interactions predict risk for post-ptca restenosis: an approach to the genetic analysis of common complex disease, Pharmacogenomics J. 2 (2002) 197–201.