Statistics and Computing manuscript No.
(will be inserted by the editor)

A novel algorithmic approach to Bayesian Logic Regression

Aliaksandr Hubin · Geir Storvik · Florian Frommlet

Received: date / Accepted: date

Abstract Logic regression was developed more than a decade ago as a tool to construct predictors from Boolean combinations of binary covariates. It has been mainly used to model epistatic effects in genetic association studies, which is very appealing due to the intuitive interpretation of logic expressions to describe the interaction between genetic variations. Nevertheless logic regression has remained less well known than other approaches to epistatic association mapping. Here we will adopt an advanced evolutionary algorithm called GMJMCMC (Genetically modified Mode Jumping Markov Chain Monte Carlo) to perform Bayesian model selection in the space of logic regression models. After describing the algorithmic details of GMJMCMC we perform a comprehensive simulation study that illustrates its performance given logic regression terms of various complexity. Specifically GMJMCMC is shown to be able to identify three-way and even four-way interactions with relatively large power, a level of complexity which has not been achieved by previous implementations of logic regression. We apply GMJMCMC to reanalyze QTL mapping data for Recombinant Inbred Lines in *Arabidopsis thaliana* and from a backcross population in *Drosophila* where we identify several interesting epistatic effects.

Keywords Logic Regression · Bayesian model averaging · Bayesian variable selection · Mode Jumping Monte Carlo Markov Chain · Genetic algorithm · Simulation · QTL mapping.

The first two authors gratefully acknowledge the financial support of the *CELS project at the University of Oslo*, http://www.mn.uio.no/math/english/research/groups/cels/index.html.

A. Hubin, G. Storvik
Department of Mathematics
University of Oslo
Moltke Moes vei 35
0851 Oslo, Norway

F. Frommlet
Department of Medical Statistics (CEMSIIS),
Medical University of Vienna
Spitalgasse 23
A-1090 Vienna, Austria
E-mail: florian.frommlet@meduniwien.ac.at

1 Introduction

Logic regression (not to be confused with logistic regression) was developed as a general tool to obtain predictive models based on Boolean combinations of binary covariates (Ruczinski et al, 2003). Its primary application area is epistatic association mapping as pioneered by Ruczinski et al (2004) and Kooperberg and Ruczinski (2005) although already early on the method was also used in other areas (Keles et al, 2004; Janes et al, 2005). Important contributions to the development of logic regression were later made by the group of Katja Ickstadt (Fritsch, 2006; Schwender and Ickstadt, 2008), which also provided a comparison of different implementations of logic regression (Fritsch and Ickstadt, 2007). Schwender and Ruczinski (2010) gave a brief introduction with various applications and potential extensions of logic regression.

Recently a systematic comparison of the performance of logic regression and a more classical regression approach based on Cockerham's coding (Wang and Zeng, 2009) to detect interactions illustrated the advantages of logic regression to detect epistasic effects in QTL mapping (Malina et al, 2014). Given the potential of logic regression to detect interpretable interaction effects in a regression setting it is rather surprising that it has not yet become wider addressed in applications.

Originally logic regression was introduced together with likelihood based model selection, where simulated anneal-

ing served as a strategy to obtain one "best" model (see Ruczinski et al, 2003, for details). However, assuming that there is one "best" model disregards the problem of model uncertainty. Whilst this approach works well in simulation studies, it seems to be quite an unrealistic assumption in real world applications, where there often is no "true" model. Hence Bayesian model averaging becomes important which implicitly takes into account model uncertainty.

Bayesian versions of logic regression combined with model exploration include Monte Carlo logic regression (MCLR) (Kooperberg and Ruczinski, 2005) and the full Bayesian version of logic regression (FBLR) by Fritsch (2006). Both MCLR and FBLR use Markov Chain Monte Carlo (MCMC) algorithms for searching through the space of models and parameters. Inference is then based on a large number of models instead of just one model as in the original version of logic regression. MCLR utilizes a geometric prior on the size of the model (defined through the number of logic terms and their complexity). All models of the same size get the same prior probability while larger models implicitly are penalized. Regression parameters are marginalized out, significantly simplifying computational complexity.

In contrast FBLR is performed on a joint space of parameters and models. FBLR uses multivariate normal priors for regression parameters, while model size is furnished with a slightly different prior serving similar purposes as the MCLR prior. In case of a large number of binary covariates these MCMC based methods might require extremely long Markov chains to guarantee convergence which can make them unfeasible in practice. Additionally both of them utilize simple Metropolis-Hastings settings which, together with the fact that the search space is often multimodal, increases the probability that they are stuck in local extrema for a significant amount of time.

In this paper we propose a new approach for Bayesian logic regression including model uncertainty. We introduce a novel prior for the topology of logic regression models which is slightly simpler to compute than the one used by MCLR and which still shows excellent properties in terms of controlling false discoveries. We consider two different priors for regression coefficients: Jeffrey's prior which corresponds to computing marginal likelihoods with the Laplace approximation as in BIC-like model selection criteria and the robust g-priors as a state of the art choice for priors of regression coefficients in variable selection problems. For the robust g-prior the marginal likelihood is efficiently computed using ILA, the integrated Laplace approximation (Li and Clyde, 2015).

The main contribution of this paper is the proposed search algorithm, named GMJMCMC, which provides a better search strategy for exploring the model space than previous approaches. GMJMCMC combines genetic algorithm ideas with the mode jumping Markov Chain Monte Carlo (MJMCMC) algorithm (Hubin and Storvik, 2016a) in order to be able to jump between local modes in the model space. After formally introducing logic regression and describing the GMJMCMC algorithm in detail we will present results from a comprehensive simulation study. The performance of GMJMCMC is compared with MCLR and FBLR in case of logistic models (binary responses) and additionally analyzed for linear models (quantitative responses). Models of different complexities are studied which allows us to illustrate the potential of GMJMCMC to detect higher order interactions. Finally we apply our logic regression approach to perform QTL mapping using two publicly available data sets. The first study is concerned with the hypocotyledonous stem length in *Arabidopsis thaliana* using Recombinant Inbred Line (RIL) data (Balasubramanian et al, 2009), the second one considers various traits from backcross data of *Drosophila Simulans* and *Drosophila Mauritana* (Zeng et al, 2000).

2 Methods

2.1 Logic regression

The method of logic regression (Ruczinski et al, 2003) was specifically designed for the situation where covariates are binary and predictors are defined as logic expressions operating on these binary variables. Logic regression can be applied in the context of the generalized linear model (GLM) as demonstrated in Malina et al (2014). It can also be easily expanded to the domain of generalized linear mixed models (GLMM), but to keep our presentation as simple as possible we will focus here on generalized linear regression models.

Consider a response variable $Y \in \mathbb{R}$, together with m binary covariates $X_1, X_2, \ldots, X_m$. Our primary example will be genetic association studies where, depending on the context, each binary covariate, $X_j, \quad j \in \{1, 2, \ldots, m\}, \quad$ can have a different interpretation. In QTL mapping with backcross design or recombinant inbred lines X_j simply codes the two possible genetic variants. In case of intercross design or in outbred populations different X_j will be used to code dominant and recessive effects (see for example Malina et al, 2014). We will adopt the usual convention that a value 1 corresponds to logical TRUE and a value 0 to logical FALSE where the immediate interpretation in our examples is that a specific marker is associated with a trait or not. Each combination of the binary variables X_j with the logical operators $\wedge$ (AND), $\vee$ (OR) and X^c (NOT X), is called a logic expression (for example $L = (X_1 \wedge X_2) \vee X_3^c$). Following the nomenclature of Kooperberg and Ruczinski (2005) we will refer to logic expressions as *trees*, whereas the primary variables contained in each tree are called *leaves*. The set of leaves of a tree L will be denoted by $v(L)$, that is for the specified example above we have $v(L) = \{X_1, X_2, X_3\}$.

We will study logic regression in the context of the generalized linear model (glm, see McCullagh and Nelder (1989)) of the form

$$Y \sim \mathfrak{f}\,(y \mid \mu(\boldsymbol{X}); \phi) \tag{1}$$

$$h\,(\mu(\boldsymbol{X})) = \alpha + \sum_{j=1}^{q} \gamma_j \beta_j L_j, \tag{2}$$

where $\mathfrak{f}$ denotes the parametric distribution of Y belonging to the exponential family with mean $\mu(\boldsymbol{X})$ and dispersion parameter ϕ. The function h is an appropriate link function, α and $\beta_j, j \in \{1, ..., q\}$ are unknown regression parameters, and γ_j is the indicator variable which specifies whether the tree L_j is included in the model. For the sake of simplicity we abbreviate by $\mu(\boldsymbol{X})$ the complex dependence of the mean μ on X via the logic expressions L_j according to (2). Our primary examples are linear regression for quantitative responses and logistic regression for dichotomous responses but the implementation of our approach works for any generalized linear model.

We will restrict ourselves to models which include no more than k_{max} trees and each tree has at most C_{max} leaves. Consequently the total number of considered trees q will be finite. The vector of binary random variables $M = (\gamma_1, \ldots, \gamma_q)$ fully characterizes a model in terms of which logical expressions are included. Here we go along with the usual convention in the context of variable selection that 'model' refers to the set of regressors and does not take into account the specific values of the non-zero regression coefficients.

2.1.1 Bayesian model specification

For a fully Bayesian approach one needs prior specifications for the model topology characterized by the index vector M as well as for the coefficients α and β_j belonging to a specific model M. We start with defining the prior for M by

$$p(M) \propto \mathbb{I}\,(|M| \leq k_{max}) \prod_{j=1}^{q} \rho(\gamma_j). \tag{3}$$

Here $|M| = \sum_{j=1}^{q} \gamma_j$ is the number of logical trees included in the model and k_{max} being the maximum number of trees allowed per model. The factors $\rho(\gamma_j)$ are introduced to give smaller prior probabilities to more complex trees. Specifically we consider

$$\rho(\gamma_j) = a^{\gamma_j c(L_j)} \tag{4}$$

with $0 < a < 1$ and $c(L_j) \geq 0$ being a non-decreasing measure for the complexity of the corresponding logical trees. In case of $\gamma_j = 0$ it holds that $\rho(\gamma_j) = 1$ and thus the prior probability for model M only consists of the product of $\rho(\gamma_j)$ for all trees included in the model. It follows that if M and M' are two vectors only differing in one component, say $\gamma'_j = 1$ and $\gamma_j = 0$, then

$$\frac{p(M')}{p(M)} = a^{c(L_j)} < 1$$

showing that larger models are penalized more. This result easily generalizes to the comparison of more different models and provides the basic intuition behind the chosen prior.

The prior choice implies a distribution for the model size $|M|$. For $k_{max} = q$ and a constant complexity value on all trees, $|M|$ follows a binomial distribution. With varying complexity measures, $|M|$ follows the *Poisson binomial* distribution (Wang, 1993) which is a unimodal distribution with $E[|M|] = \sum_{j=1}^{q} p_j$ and $\mathrm{Var}[|M|] = \sum_{j=1}^{q} p_j(1-p_j)$ where $p_j = a^{c(L_j)}/(1 + a^{c(L_j)})$. A truncated version of this distribution is obtained for $k_{max} < q$.

The choices of a and the complexity measure $c(L_j)$ are crucial for the quality of the model prior. Let $N(s)$ be the total number of trees having s leaves which will be estimated below. Choosing $a = e^{-1}$ and $c(L_j) = \log N(s_j)$ as long as the number of leaves is not larger than C_{max} results for $\gamma_j = 1$ in

$$a^{c(L_j)} = \frac{1}{N(s_j)}\,, \quad s_j \leq C_{max}\,.$$

Therefore the multiplicative contribution of a specific tree of size s to the model prior will be indirectly proportional to the total number of trees $N(s)$ having s leaves as long as $s \leq C_{max}$. Given that $N(s)$ is rapidly growing with the tree size s this choice gives smaller prior probabilities for larger trees. The resulting penalty closely resembles the Bonferroni correction in multiple testing similarly as discussed for example by Bogdan et al (2008b) in the context of modifications of the BIC.

To compute a rough approximation of $N(s)$ we ignore logic expressions including the same variable multiple times. Then there are $\binom{m}{s}$ possibilities to select variables. Each variable can undergo logic negation giving s binary choices and furthermore there are $s-1$ logic symbols $(\vee, \wedge)$ to be chosen resulting in 2^{2s-1} different expressions. However, due to De Morgan's law half of the expressions provide identical logic regression models. This gives

$$N(s) = \binom{m}{s} 2^{2s-2}. \tag{5}$$

Finally for a model of size $k = |M|$ the full model prior is of the form

$$P(M) \propto \mathbb{I}\,(k \leq k_{max}) \prod_{r=1}^{k} \frac{\mathbb{I}\,(s_{j_r} \leq C_{max})}{\binom{m}{s_{j_r}} 2^{2s_{j_r}-2}}\,, \tag{6}$$

where $j_1, \ldots, j_k$ refer to the k trees of model M.

We will next discuss priors for the parameters given a specific model M. The glm formulation (1) includes a dispersion parameter ϕ, which for example in case of the linear model is connected with the variance term σ^2 for the underlying normal distribution. If a glm has a dispersion parameter then for the sake of simplicity we will adopt the commonly used improper prior (Li and Clyde, 2015; Bayarri et al, 2012)

$$\pi(\phi) = \phi^{-1} \ . \tag{7}$$

If a glm does not include a dispersion parameter (like logistic regression) then one simply sets $\phi = 1$.

Concerning the intercept α and the regression coefficients β_j, where $j \in \{j_1, ..., j_{|M|}\}$ correspond to the non-zero coefficients of model M, we will consider two different types of priors, simple Jeffrey's priors and robust g-priors. Jeffrey's prior (Chen et al, 2008) assumes for the parameters of the model an improper prior distribution of the form

$$\pi_\alpha(\alpha)\pi_\beta(\boldsymbol{\beta}) = |J_n(\hat{\alpha}, \hat{\boldsymbol{\beta}})|^{\frac{1}{2}} \ , \tag{8}$$

where $\hat{\alpha}$ and $\hat{\boldsymbol{\beta}}$ are the maximum likelihood estimates of the coefficients. To obtain model posterior probabilities according to equation (12) one needs to evaluate the marginal likelihood of the model $P(Y \mid M)$ by integrating over all parameters of the model which is often a fairly difficult task. The greatest advantage of Jeffrey's prior is that this integration becomes rather simple due to its relationship with the Laplace approximation (Claeskens and Hjort, 2008). In case of the Gaussian model choosing Jeffrey's prior (8) for the coefficients and the simple prior (7) for the variance term yields that the Laplace approximation becomes exact (Claeskens and Hjort, 2008) and gives a marginal likelihood of the simple form

$$P(Y \mid M) \propto P(Y \mid M, \hat{\theta})\, n^{\frac{|M|}{2}} \ , \tag{9}$$

where $\hat{\theta}$ refers to the maximum likelihood estimates of all parameters involved. On the log scale this exactly corresponds to the BIC model selection criterion (Schwarz, 1978) when using a uniform model prior. In case of logistic regression the marginal likelihood under Jeffrey's prior becomes approximately (9) with an error of order $O(n^{-1})$ (Tierney and Kadane, 1986; Claeskens and Hjort, 2008). Barber et al (2016) also describe that Laplace approximations of the marginal likelihood yield very accurate results and can be trusted in Bayesian model selection problems.

Although there are many situations in which selection based on BIC like criteria works perfectly well, within the Bayesian literature using Jeffrey's prior for model selection has been widely criticized for not being consistent once the true model coincides with the null model (Bayarri et al, 2012). A large number of alternative priors have been studied, see for example Li and Clyde (2015) who give a comprehensive review on the state of the art of g-priors. In a recent paper Bayarri et al (2012) gave theoretical arguments in case of the linear model which recommend the robust g-prior, which is consistent in all situations and yields errors diminishing significantly faster than other prior choices. Thus we will introduce the robust g-prior as an alternative to Jeffrey's prior. However, we want to point out that the choice of priors for the regression coefficients is not the real focus of this paper.

Our description of robust g-priors follows Li and Clyde (2015) who consider an inproper constant prior for the intercept, $P(\alpha) \propto 1$, and a mixture g-prior for the regression coefficients $\beta_j, j \in \{j_1, ..., j_{|M|}\}$ of the form

$$P(\boldsymbol{\beta} \mid g) \sim N_{|M|}\left(\mathbf{0}, g \cdot \phi \mathcal{J}_n(\hat{\boldsymbol{\beta}})^{-1}\right) \ . \tag{10}$$

Here $\mathcal{J}_n(\hat{\boldsymbol{\beta}})$ is the observed information and g itself is assumed to be distributed according to the so called truncated Compound Confluence Hypergeometric (tCCH) prior

$$P\left(\frac{1}{1+g}\right) \sim tCCH\left(\frac{a}{2}, \frac{b}{2}, r, \frac{s}{2}, v, \kappa\right). \tag{11}$$

This family of mixtures of g-priors includes a large number of priors discussed in the literature, see Li and Clyde (2015) for more details. The recommended robust g-prior is a particular case with the following choice of parameters:

$$a = 1, b = 2, r = 1.5, s = 0, v = \frac{n+1}{|M|+1}, \kappa = 1 \ .$$

Under this prior specification precise integrated Laplace approximations of the marginal likelihood for GLM are given by Li and Clyde (2015), whilst exact values are available for Gaussian models (Li and Clyde, 2015; Bayarri et al, 2012).

2.2 Computing posterior probabilities

Given prior probabilities for any logic regression model M the model posterior probability can be computed according to Bayes formula as

$$P(M \mid Y) = \frac{P(Y \mid M)P(M)}{\sum_{M' \in \Omega} P(Y \mid M')P(M')} \ , \tag{12}$$

where $P(Y \mid M)$ denotes the integrated (or marginal) likelihood for model M and Ω is the set of all models in the model space. The sum in the denominator involves a huge number of terms and it is impossible to compute all of them. Classical MCMC based approaches (like MCLR and FBLR) overcome this problem by estimating model posteriors with the relative frequency with which a specific model M occurs in the Markov chain. In case of an ultrahigh-dimensional model space (like in case of logic regression) this is computationally extremely challenging and might require chain lengths which are prohibitive for practical applications.

An alternative approach makes use of the fact that most of the summands in the denominator of (12) will be so small that they can be neglected. Considering a subset $\Omega^* \subseteq \Omega$ containing the most important models we can therefore approximate (12) by

$$P(M \mid Y) \approx \tilde{P}(M \mid Y) = \frac{P(Y \mid M)P(M)}{\sum_{M' \in \Omega^*} P(Y \mid M')P(M')} . \tag{13}$$

To obtain good estimates we have to search in the model space for those models that contribute significantly to the sum in the denominator, that is for those models with large posterior probabilities or equivalently with large values of $P(Y \mid M)P(M)$. In Frommlet et al (2012) specific memetic algorithms were developed to perform the model search for linear regression. Here we will rely upon the GMJMCMC algorithm to be described in the next section. For now we assume that some method for computing of the marginal likelihood $P(Y \mid M)$ is available. The details of such computation depend on the prior specifications of the parameters of a particular model and are given for the particular examples in the experimental sections.

Based on model posterior probabilities one can easily obtain an estimate of the posterior probability for a logic expression L to be included in a model (also referred to as the marginal inclusion probability) by

$$\tilde{P}(L \mid Y) = \sum_{M \in \Omega^* : L \in T(M)} \tilde{P}(M \mid Y). \tag{14}$$

Inference on trees can then be performed by means of selecting those trees with a posterior probability being larger than some threshold probability π_C. More generally one can approximate the posterior probability of some parameter Δ via model averaging as

$$\tilde{P}(\Delta \mid Y) = \sum_{M \in \Omega^*} P(\Delta \mid M, Y)\tilde{P}(M \mid Y) , \tag{15}$$

where Δ might be for example the predictor of unobserved data based on a specific set of covariates.

2.3 The GMJMCMC algorithm

To fix ideas consider first a variable selection problem with q potential covariates to enter a model. Recall that γ_j needs to be 1 if the j-th variable is to be included into the model and 0 otherwise. A model M is thus specified by the vector $\boldsymbol{\gamma} = (\gamma_1, ..., \gamma_q)$ and the general model space Ω is of size 2^q. If this discrete model space is multimodal in terms of model posterior probabilities then simple MCMC algorithms typically run into problems by staying for too long in the vicinity of local maxima. Recently, the mode jumping MCMC procedure (MJMCMC) was proposed by Hubin and Storvik (2016a) to overcome this issue.

MJMCMC is a proper MCMC algorithm equipped with the possibility to jump between different modes within the discrete model space. The key to the success of MJMCMC is the generation of good proposals of models which are not too close to the current state. This is achieved by first making a large jump (changing many model components) and then performing local optimization within the discrete model space to obtain a proposal model. Within a Metropolis-Hastings setting a valid acceptance probability is then constructed using symmetric backward kernels, which guarantees that the resulting Markov chain is ergodic and has the desired limiting distribution (Hubin and Storvik, 2016a).

The MJMCMC algorithm requires that all of the covariates defining the model space are known in advance and are all considered at each iteration of the algorithm. In case of logic regression the covariates are trees and a major problem in this setting is that it is quite difficult to fully specify the space Ω. In fact it is even difficult to specify the number q of the total number of feasible trees. To solve this problem we present an adaptive algorithm called Genetically Modified MJMCMC (GMJMCMC), where MJMCMC is embedded in the iterative setting of a genetic algorithm. In each iteration only a given set $\mathcal{S}$ of trees (of fixed size d) is considered. Each $\mathcal{S}$ then induces a separate *search space* for MJMCMC. In the language of genetic algorithms $\mathcal{S}$ is the *population*, which dynamically evolves to allow MJMCMC exploring different reasonable parts of the unfeasibly large total search space. The resulting algorithm is similar to feature engineering (Xu et al, 2012) and allows to consider combinations of covariates that can be adapted throughout the search.

To be more specific, we consider different populations $\mathcal{S}_1, \mathcal{S}_2, ...$ where each $\mathcal{S}_t$ is a set of d trees. For each given population a fixed number of MJMCMC steps is performed. Since the MJMCMC algorithm is specified in full detail in Hubin and Storvik (2016a), we will concentrate here on describing the evolutionary dynamics yielding subsequent populations $\mathcal{S}_t$. In principle it is possible to construct a proper MCMC algorithm which aims at simulating from extended models of the form $P(M, \mathcal{S} \mid Y)$ having $P(M \mid Y)$ as a stationary distribution (to be published in a forthcoming paper). However, utilization of the approximation (13) in combination with exact or approximated marginal likelihoods allows us to compute posterior probabilities for all models in Ω^* which have been visited at least once by the algorithm. Consequently we do not need to fulfill detailed balance which is typically required by MCMC when model posterior probabilities are estimated by the relative frequency of how often a model has been visited.

The algorithm is initialized by first running MJMCMC for a given number of iterations N_{init} on the set of all binary covariates $X_1, ..., X_m$ as potential regressors, but not

including any interactions. The first $d_1 < d$ members of population $\mathcal{S}_1$ are then defined to be the d_1 trees with largest marginal inclusion probability. In our current implementation we select the d_1 leaves which have posterior probabilities larger than ρ_{min}, thus d_1 is not pre-specified but is obtained in a data driven way. For later reference we denote this set of d_1 leaves by $\mathcal{S}_0$. The remaining $d - d_1$ members of $\mathcal{S}_1$ are obtained by forming logic expressions from the leaves of $\mathcal{S}_0$ where trees are generated randomly by means of the crossover operation described below. In practice one first has to choose some k_{max} which will depend on the expected number of trees to enter the model in the problem one studies. The choice of d can then be guided by the results of Theorem 1 given below.

After $\mathcal{S}_1$ has been initialized MJMCMC is performed for a fixed number of iterations N_{expl} before the next population $\mathcal{S}_2$ is generated. This process is iterated for T_{max} populations $S_t, t \in \{1, ..., T_{max}\}$. The d_1 input trees from the initialization procedure remain in all populations $\mathcal{S}_t$ throughout our search. Other trees from the population $\mathcal{S}_t$ with low marginal inclusion probabilities (below a threshold ρ_{min}) will be substituted by trees which are generated by crossover, mutation and reduction operators to be described in more detail below.

Let D_t be the set of trees to be deleted from $\mathcal{S}_t$. Then $|D_t|$ replacement trees must be generated instead. Each replacement tree is generated randomly by a *crossover* operator with probability P_c and by a *mutation* operator with probability $P_m = 1 - P_c$. A *reduction* operator is applied if *mutation* or *crossover* gives a tree larger than the maximal tree size C_{max}.

Crossover: Two *parent trees* are selected from $\mathcal{S}_t$ with probabilities proportional to the approximated marginal inclusion probabilities of trees in $\mathcal{S}_t$. Then each one of the parents is inverted with probability P_{not} by the logical not ^{c} operator, before they are combined with a $\wedge$ operator with probability P_{and} and with a $\vee$ operator otherwise. Hence the crossover operator gives trees of the form $L_{j_1} \wedge L_{j_2}$ or $L_{j_1} \vee L_{j_2}$ where either L_{j_i} or $L_{j_i}^c$ is in $\mathcal{S}_t$ for $i = 1, 2$.

Mutation: One parent tree is selected from $\mathcal{S}_t$ with probability proportional to the approximated marginal inclusion probabilities of trees in $\mathcal{S}_t$, whilst the other parent tree is selected uniformly from the set of $m - d_1$ leaves which did not make it into the initial population $\mathcal{S}_0$. Then just like for the crossover operator each of the parents is inverted with probability P_{not} by the logical not ^{c} operator, before they are combined with a $\wedge$ operator with probability P_{and} and with a $\vee$ operator otherwise. The mutation operator gives trees of the form $L_{j_1} \wedge X$ or $L_{j_1} \vee X$ where either L_{j_1} or $L_{j_1}^c$ is in $\mathcal{S}_t$ and X or X^c is in D_0.

Reduction: A new tree is generated from a tree by deleting a subset of leaves, where each leave has a probability of ρ_{del} to be deleted. The pruning of the tree is performed in a natural way meaning that the 'closest' logical operators of the deleted leaves are also deleted. If the deleted leave is not on the boundaries of the original tree the operation is resulting in obtaining two separated subtrees. The resulting subtrees are then combined in a tree with a $\wedge$ operator with probability P_{and} or with a $\vee$ operator otherwise.

For all three operators it holds that if the newly generated tree is already present in $\mathcal{S}_t$ then it is not considered for $\mathcal{S}_{t+1}$ but rather a new replacement tree is proposed instead. The pseudo-code **Algorithm 1** describes the full GMJMCMC algorithm. For each iteration t the initial model for the next MJMCMC run is constructed by randomly selecting trees from $\mathcal{S}_t$ with probability P_{init}. For the final population $\mathcal{S}_{T_{max}}$, MJMCMC is run until M_{fin} unique models are visited (within $\mathcal{S}_{T_{max}}$). M_{fin} should be sufficiently large to obtain good MJMCMC based approximations of the posterior parameters of interest based on the final search space $\mathcal{S}_{T_{max}}$.

Algorithm 1 GMJMCMC

1: Run the MJMCMC algorithm for N_{init} iterations on $X_1, ..., X_m$ and define $\mathcal{S}_0$ as the set of d_1 variables among them with the largest estimated marginal inclusion probabilities.
2: Generate $d - d_1$ trees by randomly selecting crossover operations of elements from $\mathcal{S}_0$ and add those trees to the set $\mathcal{S}_0$ to obtain $\mathcal{S}_1$.
3: Run the MJMCMC algorithm within search space $\mathcal{S}_1$.
4: **for** $t = 2, ..., T_{max}$ **do**
5: Delete trees within $\mathcal{S}_{t-1} \backslash \mathcal{S}_0$ which have estimated inclusion probabilities less than ρ_{min}.
6: Add new trees which are generated by crossover, mutation or reduction operators until the having again a set of size d, which becomes $\mathcal{S}_t$.
7: Run the MJMCMC algorithm within search space $\mathcal{S}_t$.
8: **end for**

The following result is concerned with consistency of probability estimates of GMJMCMC when the number of iterations increases.

Theorem 1 *Assume Ω^* is the set of models visited through the GMJMCMC algorithm where $d - d_1 \geq k_{max}$. Then the model estimates based on* (13) *will converge to the true model probabilities as the number of iterations $T_{\max}$ converges to ∞.*

Proof Note that the approximation (13) will provide the exact answer if $\Omega^* = \Omega$. It is therefore enough to show that the algorithm in the limit will have visited all possible models. Since $\mathcal{S}_0$ is generated in the first step and never changed, we will consider it to be fixed.

Define M_{S_t} to be the last model visited by the MJMCMC algorithm on search space $\mathcal{S}_t$. Then the construction of $\mathcal{S}_{t+1}$ only depends on $(\mathcal{S}_t, M_{S_t}, \boldsymbol{X})$ while $M_{\mathcal{S}_{t+1}}$ only depends on $\mathcal{S}_{t+1}$. Therefore $\{(\mathcal{S}_t, M_{S_t}, \boldsymbol{X})\}$ is a Markov chain. Assume now $\mathcal{S}$ and $\mathcal{S}'$ are two populations differing

in one component with $L \in \mathcal{S}$, $L' \in \mathcal{S}'$, $L \neq L'$. Define L_{sub} to be any tree that is a subtree of both L and L' (where a subtree is defined as a tree which can be obtained by reduction) and $\mathcal{S}_{sub}$ to be the search space where L is substituted with L_{sub} in $\mathcal{S}$. Then it is possible to move from S to S_{sub} in l steps using first *mutations* and *crossovers* to grow a tree L^* of size larger than C_{max}, which can undergo *reduction* (note that although only trees that have low enough estimated marginal inclusion probabilities can be deleted, there will always be a positive probability that marginal inclusion probabilities are estimated to be smaller than the threshold ρ_{min}) to get to L_{sub}. Further, assuming the difference in size between L_{sub} and L' is r, a move from S_{sub} to S' can be performed by r steps of *mutations* or *crossovers*. Two search spaces which differ in s trees can be reached by s combinations of the moves described above. Since also any model within a search space can be visited, the Markov chain $\{(\mathcal{S}_t, M_{\mathcal{S}_t}, \boldsymbol{X})\}$ is irreducible. Since the state space for this Markov chain is finite, it is also recurrent, and there exists a stationary distribution with positive probabilities on every model. Thereby, all states, including all possible models of maximum size d, will eventually be visited.

When $d_1 > 0$, some restrictions on the possible search spaces are introduced. However, when $d - d_1 \geq k_{max}$, any model of maximum size k_{max} *will* eventually be visited.

Remark 1 If $d - d_1 < k_{max}$, then every model of size up to $d - d_1$ plus some of the larger models will eventually be visited, although the model space will get some additional constraints.At the same time in practice it is more important that $d - d_1 \geq k^*$, where k^* is the size of the true model. Unfortunately neither k^* nor d_1 are known in advance, and one has to make reasonable choices of k_{max} and d depending on the problem one analyses. ■

Remark 2 The result of Theorem 1 relies on exact calculation of the marginal likelihood $P(Y \mid M)$. Apart from the linear model, the calculation of $P(Y \mid M)$ is typically based on an approximation, giving similar approximations to the model probabilities. How precise these approximations are will depend on the type of method used. The current implementation includes Laplace approximations, integrated Laplace approximations, and integrated nested Laplace approximations. In principle other methods like those from Chib, or Chib and Jaliazkov could be incorporated relatively easily (Hubin and Storvik, 2016b), resulting however in longer runtimes.

Parallelization

Due to our interest in exploring as many *unique* high quality models as possible and doing it as fast as possible, running multiple parallel chains is likely to be computationally beneficial compared to running one long chain. The process can be embarrassingly parallelized into B chains using several CPUs, GPUs or clusters. If one is mainly interested in model probabilities, then equation (13) can be directly applied with Ω^* now being the set of unique models visited within all runs. However, we suggest a more memory efficient approach. If some statistic Δ is of interest, one can utilize the following posterior estimates based on weighted sums over individual runs:

$$\tilde{P}(\Delta \mid Y) = \sum_{b=1}^{B} w_b \tilde{P}_b(\Delta \mid Y) \; . \tag{16}$$

Here w_b is a set of weights which will be specified below and $\tilde{P}_b(\Delta \mid Y)$ are the posteriors obtained with formula (15) from run b of GMJMCMC.

Due to the irreducibility of the GMJMCMC procedure it holds that $\lim_{k\to\infty} \tilde{P}(\Delta \mid Y) = P(\Delta \mid Y)$ where k is the number of iterations. Thus for any set of normalized weights the approximation $\tilde{P}(\Delta \mid Y)$ converges to the true posterior probability $P(\Delta \mid Y)$. Therefore in principle any normalized set of weights w_b would work, like for example $w_b = \frac{1}{B}$. However, uniform weights have the disadvantage to potentially give too much weight to posterior estimates from chains that have not quite converged. In the following heuristic improvement w_b is chosen to be proportional to the posterior mass detected by run b,

$$w_b = \frac{\sum_{M' \in \Omega_b^*} P(Y \mid M')P(M')}{\sum_{b=1}^{B} \sum_{M' \in \Omega_b^*} P(Y \mid M')P(M')} \; .$$

This choice indirectly penalizes chains that cover smaller portions of the model space. When estimating posterior probabilities using these weights we only need, for each run, to store the following quantities: $\tilde{P}_b(\Delta \mid Y)$ for all statistics Δ of interest and $s_b = \sum_{M' \in \Omega_b^*} P(Y \mid M')P(M')$ as a *'sufficient'* statistic of the run. There is no further need of data transfer between processes.

Alternatively (as mentioned above) one might use (15) directly to approximate $P(\Delta \mid Y)$ based on the totality Ω^* of unique models explored through all of the parallel chains. This procedure might give in some cases slightly better precision than the weighted sum approach (16), but it is still only asymptotically unbiased. Moreover keeping track of all models visited by all chains requires significantly more storage in the quick memory and RAM and requires significantly more data transfers across the processes. Consequently this approach is not part of the current implementation of GMJMCMC.

The consistency result of Theorem 1 also holds in case of the suggested embarrassing parallelization. Moreover it holds that even when the number of iterations per chain is finite that letting the numbers of chains B go to infinity yields consistency of the posterior estimates as shown in Theorem

A.1 in the web supplement. The main practical consequence is that running more chains in parallel allows for having a smaller number of iterations within each thread.

Choice of algorithmic parameters Apart from the number of parallel chains, the GMJMCMC algorithm relies upon the choice of a number of parameters which were described above. Section A of the web supplement presents the values that were used in the following simulation study and in real data analysis.

3 Experiments

3.1 Simulation study

The GMJMCMC algorithm was evaluated in a simulation study divided into two parts. The first part considered three scenarios with binary responses and the second part three scenarios with quantitative responses. For each scenario we generated $N = 100$ datasets according to a regression model described by equations (1) and (2) with $n = 1000$ observations and $p = 50$ binary covariates. The covariates were assumed to be independent and were simulated for each simulation run as $X_j \sim \text{Bernoulli}(0.3)$ for $j \in \{1, \dots, 50\}$ in the first two scenarios and as $X_j \sim \text{Bernoulli}(0.5)$ for $j \in \{1, \dots, 50\}$ in the last four scenarios. All computations were performed on the Abel cluster[1].

Binary responses

The responses of the first three scenarios were sampled as Bernoulli variables with individual success probability π specified according to

$$\mathbf{S.1} : \text{logit}(\pi) = -0.7 + L_1 + L_2 + L_3$$
$$\mathbf{S.2} : \text{logit}(\pi) = -0.45 + 0.6\,L_1 + 0.6\,L_2 + 0.6\,L_3$$
$$\mathbf{S.3} : \text{logit}(\pi) = \quad 0.4 - 5\,L_1 + 9\,L_2 - 9\,L_3$$

where the corresponding logic expressions are provided in Table 1. The first two scenarios with models including only two-way interactions were copied from Fritsch (2006) except that we deliberately did not specify the trees in lexicographical order. The reason for this is that for some procedures (like stepwise search) it might be an algorithmic advantage if the effects are specified in a particular order. The second scenario is slightly more challenging than the first one due to the smaller effect sizes. The third scenario is even more demanding with a model including three-way and four-way interactions. Effect sizes were accordingly increased to give sufficient power to detect these higher order trees.

For the binary response scenarios GMJMCMC was compared with FBLR and MCLR, where GMJMCMC was run with Jeffrey's prior as well as with the robust g-prior. Additionally we ran the algorithm with Jeffrey's prior and calculated posteriors for the visited models with respect to both Jeffrey's and robust g-prior. For all three algorithms we predefined $C_{max} = 2$ leaves per tree for Scenario 1 and 2 and $C_{max} = 5$ for Scenario 3. The maximal number of trees per model was set to $k_{max} = 10$ for GMJMCMC and FBLR whereas for MCLR it is only possible to specify a maximum of $k_{max} = 5$. This is apparently due to the complexity of prior computations in MCLR. Apart from the specification of C_{max} and k_{max} we used for all 3 algorithms their default priors. In all scenarios we used $d = 15$ for the population size in GMJMCMC.

GMJMCMC was run until up to 1.6×10^6 models were visited in the first two scenarios and up to 2.7×10^6 models were visited for the third scenario (divided approximately equally on 32 parallel runs). The length of the Markov chains for FBLR and MCLR were chosen to be 2×10^6 for the first two scenarios and 3×10^6 for the third scenario.

To evaluate the performance of the different algorithms we estimated the following metrics:

Individual power - the power to detect a particular true tree (a tree from the data generating model);
Overall power - the average power over all true trees;
FP - the expected number of false positive trees;
FDR - the false discovery rate of trees;
WL - the total number of wrongly detected leaves.

Further computational details are given in Section B.1 of the web supplement.

A summary of the results for the first three simulation scenarios is provided in Table 1. In all three scenarios, MCLR performed better than FBLR, even when taking into account the positively biased summary statistics of MCLR (see Section B.1 in the web supplement). On the other hand, GMJMCMC clearly outperformed MCLR and FBLR both in terms of power and in terms of controlling the number of false positives, where using Jeffrey's prior gave slightly better results than using the robust g-prior. In the first two scenarios GMJMCMC with Jeffrey's prior worked almost perfectly. In the few instances where it did not detect the true tree it reported instead the two corresponding main effects. GMJMCMC with the robust g-prior had a few more instances where pairs of singletons were reported instead of the correct two-way interaction. FBLR and MCLR were also good at detecting the true leaves in these simple scenarios, but GMJMCMC was much better in terms of identifying the exact logical expressions.

[1] The Abel cluster node (`http://www.uio.no/english/services/it/research/hpc/abel/`) with 16 dual Intel E5-2670 (Sandy Bridge, 2.6 GHz.) CPUs and 64 GB RAM under 64 bit CentOS-6 is a shared resource for research computing.

Table 1 Results for the three simulation scenarios for binary responses. Power for individual trees, overall power, expected number of false positives (FP) and FDR are compared between FBLR, MCLR and GMJMCMC using either Jeffrey's prior (Jef.) or the robust g-prior (R.g.). All algorithms were tuned to use approximately the same computational resources. In case of MCLR we can only provide upper bounds for the power and lower bounds for FP. We also report the total number of wrongly detected leaves (WL) over all simulation runs.

	FBLR	MCLR	GMJMCMC	
Scenario 1			**Jef.**	**R. g**
$L_1 = X_1^c \wedge X_4$	0.30	≤ 0.67	0.97	0.98
$L_2 = X_5 \wedge X_9$	0.42	≤ 0.61	1.00	0.95
$L_3 = X_{11} \wedge X_8$	0.33	≤ 0.59	0.91	0.77
Overall Power	0.35	≤ 0.62	0.96	0.90
FP	3.88	≥ 2.70	0.25	0.63
FDR	0.77	≥ 0.06	0.06	0.15
WL	0	0	0	0
Scenario 2				
$L_1 = X_1^c \wedge X_4$	0.32	≤ 0.66	0.97	0.97
$L_2 = X_5 \wedge X_9$	0.40	≤ 0.67	0.99	0.96
$L_3 = X_{11} \wedge X_8$	0.37	≤ 0.60	0.86	0.76
Overall Power	0.36	≤ 0.64	0.94	0.90
FP	3.83	≥ 2.58	0.38	0.66
FDR	0.75	≥ 0.06	0.09	0.16
WL	1	1	0	0
Scenario 3				
$L_1 = X_2 \wedge X_9$	0.93	≤ 0.93	1.00	1.00
$L_2 = X_7 \wedge X_{12} \wedge X_{20}$	0.04	≤ 0.67	0.91	0.56
$L_3 = X_4 \wedge X_{10} \wedge X_{17} \wedge X_{30}$	0.00	≤ 0.19	1.00	0.56
Overall Power	0.32	≤ 0.60	0.97	0.71
FP	6.40	≥ 2.98	0.15	1.74
FDR	0.54	≥ 0.06	0.04	0.39
WL	90	72	1	0

Table 2 Results for the three simulation scenarios for linear regression. Power for individual trees, overall power, expected number of false positives (FP), FDR and the total number of wrongly detected leaves (WL) are given for parallel GMJMCMC. The four estimates in brackets for Scenario 6 are explained in the text.

Scenario 4	**Jeffrey's**	**Robust g**
$L_1 = X_5 \wedge X_9$	1.00	1.00
$L_2 = X_8 \wedge X_{11}$	0.99	1.00
$L_3 = X_1 \wedge X_4$	0.97	0.98
Overall Power	0.99	0.99
FP	0.01	0.00
FDR	0.005	0.00
WL	0	0
Scenario 5	**Jeffrey's**	**Robust g**
$L_1 = X_{37}$	1.00	1.00
$L_2 = X_2 \wedge X_9$	1.00	0.99
$L_3 = X_7 \wedge X_{12} \wedge X_{20}$	0.96	1.00
$L_4 = X_4 \wedge X_{10} \wedge X_{17} \wedge X_{30}$	0.89	0.90
Overall Power	0.96	0.97
FP	0.37	0.28
FDR	0.06	0.04
WL	2	5
Scenario 6	**Jeffrey's**	**Robust g**
$L_1 = X_7$	0.95	0.99
$L_2 = X_8$	0.98	0.99
$L_3 = X_2 \wedge X_9$	0.98	0.99
$L_4 = X_{18} \wedge X_{21}$	0.96	0.95
$L_5 = X_1 \wedge X_3 \wedge X_{27}$	1.00	1.00
$L_6 = X_{12} \wedge X_{20} \wedge X_{37}$	0.95	0.96
$L_7 = X_4 \wedge X_{10} \wedge X_{17} \wedge X_{30}$	0.32	0.45
$L_8 = X_{11} \wedge X_{13} \vee X_{19} \wedge X_{50}$	0.21 (0.93)	0.16 (0.85)
Overall Power	0.79 (0.88)	0.81 (0.90)
FP	4.28 (2.05)	4.24 (1.96)
FDR	0.38 (0.19)	0.36 (0.16)
WL	3	7

The third scenario is more complex than the previous ones but nevertheless GMJMCMC with Jeffrey's prior performed almost perfectly. GMJMCMC with the robust g-prior had more difficulties to correctly identify the three-way and four-way interaction. Both FBLR and MCLR had severe problems to detect the true logic expressions and they also reported a considerable number of wrongly detected leaves. For a more in depth discussion of these simulation results we refer to Section B.1 of the web supplement.

Finally, when the search was performed using Jeffrey's prior but the posteriors were obtained using the robust g-priors, then the posterior estimates were almost identical to those using only Jeffrey's prior throughout and there was no difference in terms of detected trees. This indicates that the choice of priors for the regression coefficients is of some importance for the quality of the search through the model space.

Continuous responses

Responses were simulated according to a Gaussian distribution with error variance $\sigma^2 = 1$ and the following three models for the expectation:

$$\mathbf{S.4}: E(Y) = 1{+}1.43\ L_1 + 0.89\ L_2 + 0.7\ L_3$$
$$\mathbf{S.5}: E(Y) = 1{+}1.5\ L_1 + 3.5\ L_2 + 9\ L_3 + 7\ L_4$$
$$\mathbf{S.6}: E(Y) = 1{+}1.5\ L_1 + 1.5\ L_2 + 6.6\ L_3 + 3.5\ L_4 + 9\ L_5 + 7\ L_6 + 7\ L_7 + 7\ L_8$$

The logic expressions used in the three different scenarios are provided in Table 2. Scenario 4 is similar to the first two scenarios for binary responses and contain only two-way interactions. The models of the last two scenarios both include trees of size 1 to 4, where scenario 5 has one tree of each size. Scenario 6 is the most complex one with two trees of each size, resulting in a model with 20 leaves in total.

For scenarios with Gaussian observations we could only study the performance of GMJMCMC since the other approaches cannot handle continuous responses (MCLR has an implementation but that does not work properly). For these scenarios the settings of GMJMCMC were adapted to the increasing complexity of the model. We used $k_{max} = 10, 10$ and 20, and $d = 15, 20$ and 40, respectively, for the

three scenarios thus allowing for models larger than twice the size of the data generating model and populations at least twice the size of the number of correct leaves involved. Furthermore, the total number of models visited by GMJMCMC before it stopped was increased to 3.5×10^6 for Scenario 6. C_{max} is set to 5 for all three of these scenarios. Otherwise all parameters of GMJMCMC were set as described for the binary responses.

Table 2 summarizes the results and further details are provided in Section B.2 of the web supplement. Scenario 4 illustrates that given a sufficiently large sample size GMJMCMC can reliably detect two-way interactions with effect sizes smaller than one standard deviation. Both Jeffrey's prior and the robust g-prior worked almost perfectly in terms of power. In this simple scenario even the type I error was almost perfectly controlled with false discovery rates equal to 0.005 for Jeffrey's prior and 0 for the robust g-prior. Interestingly the only false discovery over all 100 simulation runs was of the form $X_1 \wedge X_4 \vee X_8 \wedge X_{11}$ and is equal to $L_3 \vee L_2$. One might argue to which extent such a combination of trees should actually be counted as a false positive, a question which is further elaborated in Section B.2 of the web supplement and in the Discussion section.

The remaining two scenarios are way more complex due to the higher order interaction terms involved. In Scenario 5 the power to detect any of the four trees was very large, with only slightly smaller power for the four-way interaction. The robust g-prior had only a rather small advantage compared with Jeffrey's prior both in terms of power (overall 97% against 96%) and in terms of type I error (FDR of 4% against 6%). For both priors the majority of false positive results were connected to detecting subtrees of true trees and in all simulation runs there were only 2 wrongly detected leaves for Jeffrey's prior and 5 wrongly detected leaves for the robust g-prior.

For the last scenario we again observed large power for all true trees up to order three. For the final two expressions L_7 and L_8 of order four the results became slightly more ambiguous with power estimated to 0.32 and 0.21, respectively, for Jeffrey's prior and 0.45 and 0.16 for the robust g-prior. However, among the false positive detections we very often found the expressions $X_{11} \wedge X_{13}$, $X_{19} \wedge X_{50}$ as well as $X_{11} \wedge X_{13} \wedge X_{19} \wedge X_{50}$. In fact in 72 simulation runs for Jeffrey's prior and 69 simulation runs for the robust g-prior all of these three expressions were detected. According to the logic equivalence

$$L_8 = X_{11} \wedge X_{13} + X_{19} \wedge X_{50} - X_{11} \wedge X_{13} \wedge X_{19} \wedge X_{50}$$

one might actually consider these findings as true positives. The numbers in parentheses in Table 2 were based on taking such similarities into account, resulting in much higher power. Among the remaining false positive detections more than two thirds were subtrees of true trees or trees with misspecified logical operators but consisting of leaves corresponding to a true tree. Thus again the vast majority of false detections points towards true epistatic effects where the exact logic expression was not identified. Interestingly like in Scenario 5 GMJMCMC with the robust g-prior detected again a larger number of wrong leaves than with Jeffrey's prior.

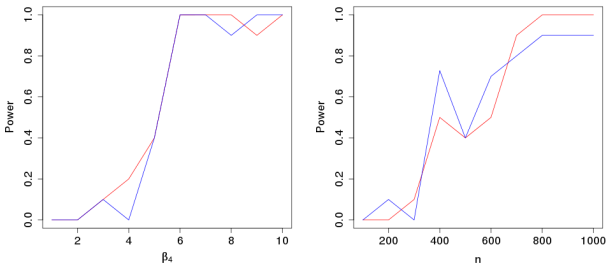

Fig. 1 Dependence of power to detect L_4 on the regression coefficient β_4 (left) and the sample size n (right) both for Jeffrey's prior (red) and the robust g-prior (blue).

Sensitivity analysis

We perform sensitivity analysis for the power to detect the four-way interaction L_4 based on $\tilde{P}(L_4|Y) > 0.5$ in Scenario 5. Specifically we consider the following three questions. How is the power effected by

1. a change in the corresponding coefficient β_4?
2. a change in the sample size n?
3. a change in the population size d?

In all three scenarios the parameters were increased uniformly in 10 steps within a given range and k_{max} was set to 20. The results presented in Figures 1-2 are based on 10 runs for each parameter value, both for Jeffrey's prior and for the robust g-prior.

The left plot of Figure 1 illustrates the dependence of power to detect L_4 on the corresponding coefficient β_4 varying between 1 and 10. For both priors the power curves sharply increase when β_4 changes from 4 to 6. This characteristic of the power curve depends on the number of leaves of the tree to be detected. Our model prior is designed to penalize more complex trees more severely in order to control FDR. For interaction terms of lower order the rise of the power curve would therefore occur already for smaller values of the corresponding regression coefficient. The fluctuations observed in the power curves in Figure 1 are due to the fairly small number of simulation runs per value.

The right plot of Figure 1 presents power curves for the detection of L_4 depending on the sample size n. Once again due to the small number of simulation runs there is some fluctuation but one can see for both priors clearly that

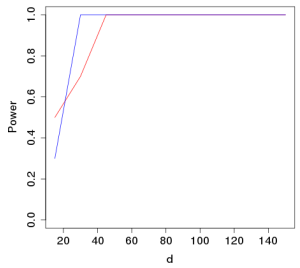

Fig. 2 Dependence of power to detect L_4 on the population size d in GMJMCMC both for Jeffrey's prior (red) and the robust g-prior (blue) for $n = 1000$.

the power grows gradually when n varies between 100 and 1000. In spite of the low resolution it is fairly clear that for an effect of $\beta_4 = 7$ one needs at least a sample size of $n = 400$ to have some power to detect this four-way interaction. One can expect that for trees of lower complexity effects of the same size can be detected already with smaller sample sizes. This is again explained by the nature of our model prior, which parsimoniously penalizes more complex trees in order to control FDR.

Figure 2 is concerned with the influence of the population size d from the GMJMCMC algorithm on the power to detect L_4. Here d ranges from 15 to 150 and $n = 1000$. As one can see for both priors power grows gradually from 0 to 1 when d changes from 15 to 45. For values of $d > 30$ the power remains stable at 1. This illustrates the statement of Theorem 1, according to which one requires $d - d_1 \geq k_{max}$ to have an irreducible algorithm in the restricted space of logic regression models. In these simulations we have $k_{max} = 20$ and $d_1 = 10$. Hence according to Theorem 1 a population size $d \geq 30$ is sufficient for asymptotic irreducibility of the GMJMCMC algorithm. For $d - d_1 < k_{max}$ irreducibility is no longer guaranteed and hence we cannot expect the approximations of the model posteriors to be precise in all cases, specifically when the model size of a data generating model is larger than $d - d_1$.

3.2 Real data analysis

Our simulation results indicate that there is no large difference in the performance of GMJMCMC between using Jeffrey's prior or the robust g-prior. On the other hand the clear computational advantage of Jeffrey's prior seems to justify to omit the robust g-prior for analyzing real data. Hence in this section GMJMCMC always refers to GMJMCMC when using Jeffrey's prior. We will analyze two data sets for QTL mapping which are publicly available. In both cases we used $k_{max} = 15$ and $d = 25$ which allows for way more complex models than we would expect to see.

Arabidopsis

Balasubramanian et al (2009) mapped several different quantitative traits (responses) in *Arabidopsis thaliana* using an advanced intercross-recombinant inbred line (RIL). Their data is publicly available as supporting information of their PLOS ONE article (Balasubramanian et al, 2009) which also gives all the details of the breeding scheme and the measurement of the different traits. We consider here only the hypocytol length in *mm* under different light conditions [2].

Genotype data is available for 220 markers distributed over the 5 chromosomes of Arabidopsis thaliana with 61, 39, 43, 31 and 46 markers, respectively. Balasubramanian et al (2009) had genotyped 224 markers but we dismissed 4 markers which had identical genotypes with other markers. The amount of missing genotype data is relatively small with a genotype rate of 93.9 % and most importantly the data contains only homozygotes (AA:49.6% vs. BB:50.4%). This means that the RIL population contains no heterozygote markers and logic regression can be directly applied using the genotype data as Boolean variables. Missing data were imputed using the R-QTL package (`http://www.rqtl.org/`).

The imputed data was then analyzed with our algorithm GMJMCMC to detect potential epistatic effects and the results are summarized in Table 3. Under blue light Balasubramanian et al (2009) reported 4 potential QTL's, the strongest one on chromosome 4 in the regions of marker X44606688 and three further fairly weak QTL on chromosomes 2, 3 and 5. Our analysis based on logic regression confirmed X44606688 and also detected those markers on chromosomes 2 and 5, though with a posterior probability slightly below 0.5. There was also some indication of a two-way interaction between the strong QTL on chromosome 4 and the QTL on chromosome 2.

Under red light the original interval mapping analysis reported the region of MSAT2.36 as a strong QTL on chromosome 2 and x44607889 as a weaker QTL on chromosome 1. Our logic regression analysis distributes the marker posterior weights on three different markers on chromosome 2 which are all in the neighborhood of MSAT2.36. Additionally there is some rather small posterior probability for an epistatic effect between this region and a marker on chromosome 1 which is somewhat close to x44607889.

Finally both for Far Red Light and for White Light our analysis essentially yielded the same results as the interval mapping analysis, when observing that under the first condition the posterior probability was again almost equally dis-

[2] Data obtained from the second to fifth column of the file `http://journals.plos.org/plosone/article/file?type=supplementary&id=info:doi/10.1371/journal.pone.0004318.s002`

Table 3 Potential additive and epistatic QTL for hypocytol length under different light conditions for Arabidopsis thaliana. Recombinant inbreed line data set taken from Balasubramanian et al (2009). Only trees for which $\tilde{P}(L \mid Y) > 0.05$ are reported.

Phenotype	Chr	Marker expression	$\tilde{P}(L \mid Y)$
Blue Light	4	X44606688	0.767
Blue Light	5	X44607250	0.335
Blue Light	2	X21607656	0.309
Blue Light	4∧2	X44606688∧X44606810	0.203
Red Light	2	MSAT2.36	0.441
Red Light	2	PHYB	0.353
Red Light	2∧1	PHYB^c∧X44606541	0.112
Red Light	2	X21607013	0.092
Far Red Light	4	MSAT4.37	0.302
Far Red Light	4	NGA1107	0.302
White Light	5	X44606159	0.632
White Light	1	X21607165	0.427

tributed between the neighboring markers MSAT4.37 and NGA1107.

In summary the sample size in this data set might be slightly too small to detect epistatic effects, although under the first two light conditions there was at least some indication for a two-way interaction.

Drosophila

As a second real data example we considered the *Drosophila* back cross data from Zeng et al (2000) [3]. There are five quantitative traits available for each species (abbreviated as `pc1`, `adjpc1`, `area`, `areat` and `tibia`) which quantify the size and shape of the posterior lobe of the male genital arch. The original publication (Zeng et al, 2000) only includes results on the first measure `pc1`, which was later analyzed for epistatic effects using a model selection approach based on the Cockerham coding (Bogdan et al, 2008a).

Compared with the Arabidopsis example this backcross data set has a much larger sample size combined with a smaller number of genetic markers, which both helps to increase the power to detect QTL. Genotype data from 45 markers is available for 471 samples from *Drosophila Simulans* and 491 samples from *Drosophila Mauritana*. Six markers are located on chromosome X, 16 markers on chromosome 2 and 23 markers on chromosome 3. Imputation of the few missing genotypes was performed by a simple maximum likelihood approach based on flanking markers. More details on the experiments and the measured traits can be found in Zeng et al (2000).

Table 4 reports trees with posterior probabilities larger than 0.3 for the trait `pc1` of *Drosophila Simulans* and compares with the model obtained with mBIC - based forward selection by Bogdan et al (2008a). The logic regression approach detected most of the main effects also previously reported, which in itself is quite interesting because as we allowed for higher order interactions we looked at a much larger model space and used therefore implicitly larger penalties than mBIC. In two locations GMJMCMC preferred a neighboring marker (*cg* instead of *gpdh* on chromosome 2 and *hb* instead of *rox* on chromosome 3. In one region on chromosome 3 mBIC selected 2 markers (*rdg*, *ninaE*) whereas GMJMCMC selected only one marker in the middle. These kind of discrepancies are quite natural due to marker correlations in back cross data (Bogdan et al, 2008a). Just like with the mBIC approach we detected a two-way interaction between chromosome 2 and chromosome 3, where on both locations the two methods chose neighboring markers, respectively. Otherwise the epistatic effect detected with both methods is identical.

Table 4 Results for *Drosophila Simulans* are presented for the trait `pc1` from Zeng et al (2000). Posterior probabilities for additive and epistatic effects detected with GMJMCMC (column $\tilde{P}(L \mid Y)$) are compared with the findings reported by Bogdan et al (2008a) using mBIC as a selection criterion (column mBIC). Posterior probabilities are only reported for trees with $\tilde{P}(L \mid Y) > 0.3$ are reported.

Marker	Chr	Marker name	$\tilde{P}(L \mid Y)$	mBIC
m2	X	w	1.000	x
m4	X	v	1.000	x
m7	2	gl	0.960	x
m9	2	cg	1.000	
m10	2	gpdh		x
m14	2	mhc	1.000	x
m18	2	sli	0.414	x
m22	2	zip	0.838	x
m23	2	lsp	0.998	x
m26	3	dbi	1.000	x
m29	3	fz	1.000	x
m32	3	rdg		x
m33	3	ht	1.000	
m35	3	ninaE		x
m37	3	mst	1.000	x
m40	3	hb	0.942	
m41	3	rox		x
m44	3	jan	1.000	x
m12, m34	2, 3	glt∧ant		x
m11, m35	2, 3	ninaE ∧ ninaC	0.998	

Table 5 contains the corresponding results for Drosophila Mauritana. As before GMJMCMC detects most of the additive effects that were reported by mBIC, though it sometimes chooses flanking markers (*ve* and *dbi* instead of *acr*, *tub* instead of *hb*). Interestingly the marker *ewg* on the X-chromosome is not reported as a main effect but rather as a two-way interaction together with *v* also on the X-chromosome, which also shows up as an additive effect. On the other hand the two-way interactions obtained with mBIC are not confirmed. Instead of the interaction between *fz* and *hb* GMJMCMC reports additional main effects on *fz* and *rox* (the neigh-

[3] Data downloaded from `ftp://statgen.ncsu.edu/pub/qtlcart/data/zengetal99`. There one can also find a linkage map in centiMorgan for the markers on three different chromosomes

Table 5 Results for *Drosophila Mauritana* are presented for the trait `pc1` from Zeng et al (2000). Posterior probabilities for additive and epistatic effects detected with GMJMCMC (column $\tilde{P}(L \mid Y)$) are compared with the findings reported by Bogdan et al (2008a) using mBIC as a selection criterion (column mBIC). Posterior probabilities are only reported for trees with $\tilde{P}(L \mid Y) > 0.3$ are reported.

Marker	Chr	Marker name	$\tilde{P}(L \mid Y)$	mBIC
m1	X	ewg		x
m4	X	v	0.994	x
m9	2	cg	1.000	x
m11	2	ninaC	0.382	x
m15	2	ddc	1.000	x
m18	2	sli	0.523	x
m22	2	zip	1.000	x
m24	3	ve	0.966	
m25	3	acr		x
m26	3	dbi	0.995	
m28	3	cyc	0.398	x
m29	3	fz	0.834	
m34	3	ant	1.000	x
m37	3	mst		x
m39	3	tub	0.999	
m40	3	hb		x
m41	3	rox	0.420	
m44	3	jan	1.000	x
m1, m2	X, X	w∨ewg	0.855	
m2, m36	X, 3	w∨fas		x
m29, m40	3, 3	fz∨hb		x

bor of *hb*). For the interaction between *w* and *fas* there are no substitutes detected.

The results for the other four traits (*adjpc1, area, areat* and *tibia*) are provided in Section C of the web supplement. In case of *Drosophila Simulans* we detect three two-way interactions for *adjpc1*. For *Drosophila mauritiana* further two-way interactions are found; two for *adjpc1*, three for *area*, and two more for *areat*. We did not find higher order interactions for any of these traits and based on the experience from our simulation study we might conclude that there are actually at least no strong higher epistatic effects.

4 Discussion

We have introduced GMJMCMC as a novel algorithm to perform Bayesian logic regression and compared it with the two existing methods MCLR (Kooperberg and Ruczinski, 2005) and FBLR (Fritsch, 2006). The main advantage of GMJMCMC is that it is designed to identify more complex logic expressions than its predecessors. Our approach differs both in terms of prior assumptions and in algorithmic details. Concerning the prior of regression coefficients we compared the simple Jeffrey's prior with the robust g-prior. Jeffrey's prior in combination with the Laplace approximation coincides with a BIC-like approximation of the marginal likelihood, which was also used by MCLR. The robust g-prior has some very appealing theoretical properties for the linear model. However, in our simulation study it gave only slightly better results than Jeffrey's prior for the linear model and in case of logistic regression actually performed worse in terms of power to detect the trees of the data generating logic regression model. However, when the search was performed using Jeffrey's prior but the posteriors were calculated with both Jeffrey's and the robust g-prior, then the results were almost identical between both priors.

With respect to the model topology we chose a prior which is somewhat similar to the one suggested by Fritsch (2006) for FBLR, but instead of using a truncated geometric prior for the number of leaves of a tree we suggest a prior which penalizes the complexity of a tree indirectly proportionally to the total number of trees of a given size. The motivation behind this prior is to control the numer of false positive detections of trees in a similar way to how the Bonferroni correction works in multiple testing.

GMJMCMC has the capacity to explore a much larger model search space than MCLR and FBLR because it manages to efficiently resolve the issue of not getting stuck in local extrema, a problem that both MCLR and FBLR have in common. In logic regression the marginal posterior probability function is typically multi-modal in the space of models, with a large number of extrema which are often rather sparsely located. Additionally, the search space for logic regression is extremely large, where even computing the total number of models is a sophisticated task. As discussed in more detail in Hubin and Storvik (2016a), in such a setting simple MCMC algorithms often get stuck in local extrema, which significantly slows down their performance and convergence might only be reached after run times which are infeasible in practice.

The success of GMJMCMC relies upon resolving the local extrema issue, which is mainly achieved by combining the following two ideas. First, when iterating through a fixed search space S, GMJMCMC utilizes the MJMCMC algorithm (Hubin and Storvik, 2016a) which was specifically constructed to explore multi-modal regression spaces efficiently. Second, the evolution of the search spaces is governed within the framework of a genetic algorithm where a population consists of a finite number of trees forming the current search space. The population is updated by discarding trees with low estimated marginal posterior probability and generating new trees with a probability depending on the approximations of marginal inclusion probabilities from the current search space. The aim of the genetic algorithm is to converge towards a population which includes the most important trees. Finally the performance of GMJMCMC is additionally boosted by running it in parallel with different starting points.

Irreducibility of the proposals both for search spaces and for models within the search spaces guarantees that asymptotically the whole model space will be explored by GMJMCMC and global extrema will at some point be reached un-

der some weak regularity conditions. Clearly the genetic algorithm used to update search spaces results in a Markov chain of model spaces. In the future it will be interesting to generalize the mode jumping ideas from Hubin and Storvik (2016a) to the Markov chain of search spaces, making it converge to the right limiting distribution in the joint space of models, parameters and search spaces, whilst remaining the property of not getting stuck in local modes.

One important question in the context of logic regression is concerned with how to define true positive and false positive detections in simulations. We adopted a rather strict point of view which might be called an 'exact tree approach': Only those detected logic expressions which were logically equivalent with trees from the data generating model were counted as true positives. While this seems to be a natural definition there are certain pitfalls and ambiguities that occur in logic regressions which might speak against this strict definition. Apart from the more obvious logic equivalences according to Boolean algebra, for example due to De Morgan's laws or the distributive law, there can be slightly more hidden logic identities in logic regression. For example the expressions $(X_1 \vee X_2) - X_1$ and $X_2 - (X_1 \wedge X_2)$ give identical models. We have seen a less trivial example including four-way interactions in Scenario 6 of our simulation study, where the data generating tree L_8 is equivalent to the expression $X_{11} \wedge X_{13} + X_{19} \wedge X_{50} - X_{11} \wedge X_{13} \wedge X_{19} \wedge X_{50}$ consisting of three trees. Furthermore, different logic expressions can be highly correlated even when they are not exactly identical.

Especially the results from the most complex Scenario 6 impose the question whether the exact tree approach is slightly too strict to define false positives. Subtrees of true trees give valuable information even if they are not describing the exact interaction. Often combinations of several subtrees and trees with misspecified logical operators can give expressions which are very close to the correct interaction term. For Scenario 6 we reported two possible summaries of the simulation results, one based strictly on the exact tree approach and the other one counting simultaneous detections of $X_{11} \wedge X_{13}$, $X_{19} \wedge X_{50}$ and $X_{11} \wedge X_{13} \wedge X_{19} \wedge X_{50}$ also as true positives. This was slightly ad hoc and we believe that good reporting of logic regression results is an area which needs further research. The output of MCLR takes a step in that direction, where only the leaves of trees are reported and if a tree has been detected then also all its subtrees are reported. However, in our opinion MCLR throws away too much information. We believe that several different layers of reporting might be more desirable, for example the exact tree approach, the MCLR approach and then something in between which does not reduce trees completely to their set of leaves. We have started to think more systematically in that direction and leave this topic open for another publication.

Our simulation study demonstrated the potential of the GMJMCMC algorithm to find true logical expressions with high power and low false discovery rate, whilst in the real data examples GMJMCMC could find interesting epistatic effects in QTL analysis. However, the current implementation has a slight tendency to prefer a set of several simple trees over a single complicated tree. Specifically it does not properly take into account that a complex tree can be represented in several equivalent ways which leaves space for further improvements. In the future we would also like to extend GMJMCMC to more general non-linear regression settings.

The R package implementing both MJMCMC and GMJMCMC is freely available on GitHub at `http://aliaksah.github.io/EMJMCMC2016/`, where one can also find examples of further logic regression applications.

References

Balasubramanian S, Schwartz C, Singh A, Warthmann N, Kim M, Maloof J, Loudet O, Trainer G, Dabi T, Borevitz J, Chory J, Weigel D (2009) QTL mapping in new Arabidopsis thaliana advanced intercross-recombinant inbred lines. PLoS One 4(2)

Barber RF, Drton M, Tan KM (2016) Laplace Approximation in High-Dimensional Bayesian Regression, Springer International Publishing, Cham, pp 15–36

Bayarri MJ, Berger JO, Forte A, García-Donato G, et al (2012) Criteria for bayesian model choice with application to variable selection. The Annals of statistics 40(3):1550–1577

Bogdan M, Frommlet F, Biecek P, Cheng R, Ghosh JK, Doerge RW (2008a) Extending the Modified Bayesian Information Criterion (mBIC) to Dense Markers and Multiple Interval Mapping. Biometrics 64(4):1162–1169

Bogdan M, Ghosh JK, Tokdar ST (2008b) A comparison of the Simes-Benjamini-Hochberg procedure with some Bayesian rules for multiple testing. IMS Collections, **Vol1**, Beyond Parametrics in Interdisciplinary Research: Fetschrift in Honor of Professor Pranab K Sen, edited by N Balakrishnan, Edsel Peña and Mervyn J Silvapulle pp 211–230

Chen MH, Ibrahim JG, Kim S (2008) Properties and implementation of jeffreys's prior in binomial regression models. Journal of the American Statistical Association 103(484):1659–1664, URL `http://www.jstor.org/stable/27640213`

Claeskens G, Hjort NL (2008) Model Selection and Model Averaging. Cambridge Series in Statistical and Probabilistic Mathematics, Cambridge University Press, DOI 10.1017/CBO9780511790485

Fritsch A (2006) A Full Bayesian Version of Logic regression for SNP Data. PhD thesis, Diploma Thesis

Fritsch A, Ickstadt K (2007) Comparing Logic Regression Based Methods for Identifying SNP Interactions. Springer Berlin / Heidelberg, Lecture Notes in Computer Science 4414:90–103

Frommlet F, Ljubic I, Arnardottir H, Bogdan M (2012) QTL Mapping Using a Memetic Algorithm with modifications of BIC as fitness function. Statistical Applications in Genetics and Molecular Biology 11(4):Article 2

Hubin A, Storvik G (2016a) Efficient mode jumping MCMC for Bayesian variable selection in GLMM. arXiv preprint arXiv:160406398

Hubin A, Storvik G (2016b) Estimating the marginal likelihood with Integrated nested Laplace approximation (INLA). ArXiv:1611.01450v1, 1611.01450

Janes H, Pepe M, Kooperberg C, Newcomb P (2005) Identifying target populations for screening or not screening using logic regression. Statistics in Medicine 24:1321–1338

Keles S, van der Laan M, Vulpe C (2004) Regulatory motif finding by logic regression. Bioinformatics 20:2799–2811

Kooperberg C, Ruczinski I (2005) Identifying Interacting SNPs Using Monte Carlo Logic Regression. Genetic Epidemiology 28:157–170

Li Y, Clyde MA (2015) Mixtures of g-priors in generalized linear models. arXiv preprint arXiv:150306913

Malina M, Ickstadt K, Schwender H, Posch M, Bogdan M (2014) Detection of epistatic effects with logic regression and a classical linear regression model. Statistical Applications in Genetics and Molecular Biology 13(1):83–104

McCullagh P, Nelder J (1989) Generalized Linear Models. 2nd Edition. Chapman and Hall, London

Ruczinski I, Kooperberg C, LeBlanc M (2003) Logic regression. J Comput Graphical Statist 12(3):474–511

Ruczinski I, Kooperberg C, LeBlanc M (2004) Exploring Interactions in High-Dimensional Genomic Data: An Overview of Logic Regression, with Applications. Journal of Multivariate Analysis 90:178–195

Schwarz G (1978) Estimating the dimension of a model. The Annals of Statistics 6:461–464

Schwender H, Ickstadt K (2008) Identification of SNP interactions using logic regression. Biostatistics 9:187–198

Schwender H, Ruczinski I (2010) Logic Regression and Its Extensions. Advances in Genetics 72:25–45

Tierney L, Kadane JB (1986) Accurate approximations for posterior moments and marginal densities. JASA 81(393):82–86, DOI 10.1080/01621459.1986.10478240

Wang T, Zeng ZB (2009) Contribution of genetic effects to genetic variance components with epistasis and linkage disequilibrium. BMC Genetics 10(1):52, DOI 10.1186/1471-2156-10-52

Wang YH (1993) On the number of successes in independent trials. Statistica Sinica pp 295–312

Xu Y, Hong K, Tsujii J, Chang EIC (2012) Feature engineering combined with machine learning and rule-based methods for structured information extraction from narrative clinical discharge summaries. Journal of the American Medical Informatics Association 19(5):824, DOI 10.1136/amiajnl-2011-000776

Zeng ZB, Liu J, Stam LF, Kao CH, Mercer JM, Laurie CC (2000) Genetic architecture of a morphological shape difference between two Drosophila species. Genetics 154:299–310