

Supplementary Material for: A novel algorithmic approach to Bayesian Logic Regression

A. HUBIN, G. STORVIK, F. FROMMLET

A GMJMCMC Algorithm

A.1 Tuning parameters

In all the simulations and in real data analysis we used the default tuning parameters of the implementation of MJMCMC downloaded from <http://aliaksah.github.io/EMJMCMC2016/>. The values which were used in the different simulation scenarios and for real data analysis for the parameters not related to MJMCMC but rather to the genetic algorithm part are presented in Table A.1.

Table A.1. Tuning parameters of GMJMCMC in the different examples (Ex.), where simple digits refer to the simulation scenario, RD1 refers to the Arabidopsis data analysis and RD2 to the Drosophila data analysis; Threads (Th.) - the number of CPUs utilized within the examples; N_{init} - the number of steps of MJMCMC during initialization; N_{expl} - the number of steps of MJMCMC between changes of population; M_{fin} - the number of unique models visited by MJMCMC for the final population; T_{max} - index of the final population; ρ_{min} - threshold for the trees to be deleted; P_{and} - probability of an *and* operator in crossovers and mutations; P_{not} - probability of using logical *not* in crossovers and mutations; P_c - probability of crossover to propose replacement trees; P_{init} - probability for a tree to be included into the initial solution for a new MJMCMC run in any iteration $t \geq 1$; ρ_{del} - probability of deletion in the reduction operator; C_{max} - maximal tree size allowed; k_{max} - maximal number of trees allowed in a model; d - size of population of genetic algorithm (number of trees searched by MJMCMC in each iteration).

Ex.	Th.	N_{init}	N_{expl}	M_{fin}	T_{max}	ρ_{min}	P_{and}	P_{not}	P_{init}	P_c	ρ_{del}	C_{max}	k_{max}	d
1	32	300	300	10000	16	0.2	1.0	0.2	0.5	0.9	0.5	2	10	15
2	32	300	300	10000	16	0.2	1.0	0.2	0.5	0.9	0.5	2	10	15
3	32	300	300	15000	33	0.2	0.9	0.1	0.5	0.9	0.5	5	10	15
4	32	300	300	10000	33	0.2	0.9	0.1	0.5	0.9	0.5	5	10	15
5	32	300	300	10000	33	0.2	0.9	0.1	0.5	0.9	0.5	5	10	20
6	32	250	250	20000	40	0.2	0.7	0.1	0.5	0.9	0.5	5	20	40
RD1	64	250	250	35000	40	0.2	0.7	0.1	0.5	0.9	0.5	5	15	25
RD2	64	250	250	15000	40	0.2	0.7	0.1	0.5	0.9	0.5	5	15	25

A.2 Theorem for parallel version of GMJMCMC

The following Theorem generalizes Theorem 1 from the manuscript to the parallelized version of GMJMCMC. Apart from letting the number of iterations go to infinity it is also possible to have only a finite number of iterations within each run but let the number of parallel runs go to infinity.

Theorem A.1. *Assume that we are running GMJMCMC in B parallel chains as describes in Section 2.3 of the manuscript. When the number of iterations within each chain b converges to infinity, the posterior estimates $\tilde{P}(\Delta | Y)$ of (16) from the manuscript will converge to $P(\Delta | Y)$.*

Assuming the search space \mathcal{S}_1 is selected randomly within the total set of possible search spaces and for a finite number of iterations within each chain b , the posterior estimates (16) will converge to $P(\Delta | Y)$ when $B \rightarrow \infty$.

Proof. When the number of iterations within each chain b converges to infinity, each $\tilde{P}_b(\Delta | Y)$ will converge to $P(\Delta | Y)$ according to Theorem 1 of the manuscript. Further, each $w_b \rightarrow 1/B$, proving the first part of the result.

When the initial search space \mathcal{S}_1 is selected randomly, any possible tree can be included. According to the construction of the initial model for the first MJMCMC run any model will have positive probability of being selected, giving the result directly. \square

Remark Selecting the search space \mathcal{S}_1 randomly among all possible models is in principle not easy due to the difficulty of specifying the complete model space. However, running the GMJMCMC algorithm with no data can be performed extremely fast, making it possible to select the initial population randomly.

B Details of Simulation Results

In this section we present further information on the simulation results of our six scenarios.

B.1 Binary Response

In case of GMJMCMC and FBLR a tree was counted as detected if its corresponding posterior probability was larger than 0.5. The power to detect a true tree is estimated by the percentage of simulation runs in which it was detected. The overall power is then defined as the average power over all individual true trees. A detected tree was counted as true positive if it was logically equivalent to a tree from the data generating model or to its logical complement, otherwise it was counted as false positive. FP denotes the average, over simulation runs, number of false positive detections and FDR was estimated as the average (over simulation runs) proportion of false discoveries, where this proportion was defined to be zero if there were no detections at all. WL is the number of binary covariates (leaves) which were not part of the data generating model but part of at least one detected tree.

Unfortunately, the output delivered by MCLR does not allow to compute the performance measures in the same way. Whenever MCLR detects a tree of size s then all subtrees are also reported as being detected. Furthermore MCLR reports for each detected tree only the set of leaves $v(L)$ and not the exact logical expression L itself. Thus it becomes impossible to define true positives by comparing the reported trees directly with the trees from the data generating model. Instead we considered for MCLR a reported tree L as a true positive whenever $v(L)$ coincided with the set of leaves of a true tree. This definition only gives an upper bound for the achieved power and is strongly biased in favor of MCLR. For the same reason, any reported tree that was a subtree of a true tree was not considered to be a false positive, resulting in only lower bounds of FP and FDR which are again strongly biased in favor of MCLR.

Table B.1 gives details about the frequencies of trees detected by the different methods. The first three lines give for each scenario the frequency with which the three true trees L_j in each scenario were detected. All further detected trees are per definition false positives.

However, we considered different classes of false positives. The first class of false positives are trees which are comprised exclusively of leaves from a true tree L_j , typically subtrees or trees with a different logic expression. Based on the output of MCLR it is not possible to determine the frequencies of this kind of false positive detections as we will discuss in the next paragraph. In case of FBLR and GMJMCMC Table B.1 provides the frequency of this kind of trees in the rows labeled $v(L_j)$. For Scenario 1 and 2 we actually provide more detailed information. Here all true trees are of size 2 and almost all detected trees of the class $v(L_j)$ consisted of single leaves (the only exception was two instances of the expression $X_8^c \wedge X_{11}$ in Scenario 1 for FBLR). We therefore explicitly present the number of detections of the first leaf and the second leaf of L_j . The $v(M)$ rows give the number of trees combining leaves from different true trees. Finally the rows $WL(s)$ are concerned with the number of trees which include s leaves which were not in the data generating model at all.

In case of MCLR the same sort of classification is not possible due to the fact that MCLR does not report the exact logical tree L that it detects but only the corresponding set of leaves $v(L)$. Furthermore MCLR automatically reports the set of leaves for all subtrees of any detected tree which makes an assessment on how often these subtrees were actually detected by MCLR impossible. As a consequence we simply discarded reported subtrees when computing summary statistics, with one exception. In case of Scenario 3 MCLR reports 40 supertrees (trees for which a tree of interest is a subtree) of L_1 , which we classified as false positives themselves but which in principle play an important role for the determination of the power to detect L_1 . We ignored the fact that for any detected supertree of L_1 MCLR automatically also reports L_1 itself as detected and we pretended that in all these cases MCLR would actually have detected L_1 itself. Another peculiarity of MCLR is that it allows to search for trees of size 4, but that it does not report if it detected any such trees. In case of the four-way interaction L_3 from Scenario 3 there were 19 simulation runs where MCLR reported all four subtrees of size 3 from L_3 and we counted those instances as true positives, although MCLR did not really report the correct four-way interaction. For Scenario 1 and 2 none of these problems with supertrees occurred for MCLR because we restricted the search to trees of maximal size 2 in accordance with the data generating model.

The first two scenarios include only two-way interactions and we observe that GMJMCMC with Jeffrey's prior worked almost perfectly well. In the few instances where it did not detect the correct tree it reported instead the two corresponding main effects, resulting in a total of 25 and 38 false positive trees for the two scenarios (corresponding to an average of 0.25 and 0.38 false positives within each simulation, see Table 1 of the main manuscript). The robust g-prior resulted in a few more false positives which were also all just single leaves instead of the two-way interactions. FBLR chose in almost two thirds of the simulation runs two main effects instead of the correct interactions. The majority of the remaining false positives combined leaves from different true trees but there was also for each scenario one expression with a wrongly detected leaf, respectively. In contrast MCLR reported in approximately two thirds of the cases trees with the correct leaves resulting in larger power than for FBLR. On the other hand MCLR reported a much larger number of trees which combined leaves from different true trees than FBLR. MCLR reported only one tree with a wrong leaf in Scenario 2 and no such tree in Scenario 1. In summary we conclude that all three methods were doing extremely well in detecting the correct leaves in these simple scenarios but GMJMCMC was better than FBLR and MCLR in identifying the exact logical expressions.

The conclusion above is even more pronounced in the third scenario, which is more complex than the previous scenarios but still allows GMJMCMC with Jeffrey's prior to perform almost perfectly. It detected both the two-way interaction L_1 and the four-way interaction L_3 with a power of 100%, and had only some minor difficulties to detect the three-way interaction L_2 . From the 15 false positive detections the majority consisted of subtrees of L_2 reported in those simulation runs where L_2 itself was not detected.

Table B.1. Number of true and false positive trees for the three simulation scenarios with a binary response. A detailed description of the different classes of false positives ($v(L_j)$, $v(M)$, $WL(s)$) is given in the text above. The columns Jef. and Rob. g correspond to GMJMCMC with Jeffrey’s prior and with the Robust g-prior, respectively.

	FBLR	MCLR	Jef.	Rob. g
S.1				
L_1	30	67	97	98
L_2	42	61	100	95
L_3	33	59	91	77
$v(L_1)$	68+69	*	3 + 3	2 + 2
$v(L_2)$	54+53	*	1 + 0	5 + 5
$v(L_3)$	60+59(+2)	*	9 + 9	25 + 24
$v(M)$	22	270	0	0
$WL(1)$	1	0	0	0
S.2				
L_1	32	66	97	97
L_2	40	67	99	96
L_3	37	60	86	76
$v(L_1)$	64 + 66	*	3 + 3	3 + 3
$v(L_2)$	56 + 60	*	1 + 1	4 + 4
$v(L_3)$	56 + 56	*	15 + 15	26 + 26
$v(M)$	24	256	0	0
$WL(1)$	1	1	0	0
S.3				
L_1	93	93 (40SupT)	100	100
L_2	4	67	91	56
L_3	0	19 (SubT)	100	56
$v(L_1)$	20	*	0	0
$v(L_2)$	162	*	8	81
$v(L_3)$	233	*	1	87
$v(M)$	167	195	5	6
$WL(1)$	34	54	1	0
$WL(2)$	16	9	0	0
$WL(3)$	8	0	0	0

Five trees were combinations of leaves from different true trees and there was only one tree including a leave which was not part of the data generating model. GMJMCMC with the robust g-prior had substantially lower power to detect L_2 and L_3 but instead reported many corresponding subtrees. There were six reported logic expressions which mixed leaves from L_2 and L_3 . In comparison, both MCLR and FBLR performed much worse and only managed to detect L_1 with fairly large power. FBLR completely failed to detect the higher order terms L_2 and L_3 whereas MCLR had at least some power to detect the three-way interaction L_2 . Both approaches reported way more false positive trees than GMJMCMC.

For FBLR we can discuss the structure of false positive detections in more detail. A large number of false positive expression were comprised of leaves from single true trees, 20 for $v(L_1)$, 162 for $v(L_2)$ and 233 for $v(L_3)$. These expressions were either subtrees of true trees or trees with misspecified logical operators and can be seen as substitutes for the true trees. Furthermore there were 167 logical expressions which combined leaves from different trees. Additionally, FBLR reported 34 trees with one wrongly detected leave, 16 trees with two wrongly detected leaves and even 8 trees of size three for which all leaves were

not part of the data generating model. Thus apart from having problems with determining the exact form of the logical expressions in this scenario FBLR produced also a large number of false positive trees which have nothing to do with the correct model at all.

The performance of MCLR was only a little bit better. With respect to the results presented in Table 1 of the main manuscript it is now even more important than for the first two scenarios to emphasize that we are dealing with upper bounds of the power and lower bounds of the number of false positives. MCLR automatically reports all subtrees of any detected tree which makes an assessment how often these tree were actually detected by MCLR impossible. As a consequence we simply discarded reported subtrees from further statistical analysis, with one exception. In case of Scenario 3 MCLR reports some supertrees of L_1 , which we classified as false positives themselves but which in principle played an important role for the determination of the power of L_1 . We ignored the fact that for any detected supertree of L_1 MCLR automatically also reports L_1 itself as detected and pretend that in all these cases MCLR would actually have detected L_1 itself. Another peculiarity of MCLR is that it allows to search for trees of size 4, but that it does not report if it detected any such trees. In case of the four-way interaction L_3 there were 19 simulation runs where MCLR reported all four subtrees of size 3 from L_4 and we counted those instances as true positives, although MCLR did not really report the correct four-way interaction. For Scenario 1 and 2 none of these problems with supertrees occurred for MCLR because we restricted the search to trees of maximal size 2 in accordance with the data generating trees.

Not counting any subtrees of reported trees as false positives gives MCLR a huge advantage, nevertheless it reported almost 20 times more false positive expressions than GMJMCMC. Among those were 40 supertrees of L_1 , which all contributed to the power of L_1 although it is not guaranteed that in all corresponding simulation runs L_1 itself was actually detected. There were 195 false positive trees which combined leaves from different true trees. It was more problematic that there were 54 trees with one wrongly detected leaf and 9 trees with two wrongly detected leaves. While there were not as many trees which were completely wrong as for FBLR there were still a considerable number of leaves reported by MCLR which were not part of the data generating model.

B.2 Continuous Response

Table B.2 gives detailed results about the frequencies of detected trees similarly to Table B.1 but now for the three linear regression scenarios. At the beginning we have again for each scenario the number of true positives with L_j referring to the trees of the data generating model. As described above we split the detections of false positives again in the classes $v(L_j)$ which refers to logic expressions consisting only of leaves from L_j , $v(M)$ which refers to logic expressions consisting of leaves from the data generating model but mixing leaves from different trees, and $WL(1)$ corresponding to trees including one wrong leaf. For the last expression of Scenario 6 it holds that $L_8 = X_{11} \wedge X_{13} + X_{19} \wedge X_{50} - X_{11} \wedge X_{13} \wedge X_{19} \wedge X_{50}$ and it turned out that in many simulation runs GMJMCMC was detecting the three expression from the alternative version. In the main manuscript we considered these findings potentially as true positives and in Table B.2 we explicitly report the frequency of detection for each of these trees.

There is not much to be said about Scenario 4 apart from the fact that the only false positive detection $L_2 \vee L_3$ was very close to the expression $L_2 + L_3$ of the data generating model. In Scenario 5 the results using Jeffrey's prior and the robust g-prior are very similar. For those trees which were detected in all simulation runs (L_1 and L_2 for Jeffrey's, L_1 and L_3 for the robust g-prior) no false positive subtrees were reported. The majority of false positives for both priors is comprised of subtrees and there are only a very small number of detections which combine leaves from two different true trees (1 for Jeffrey's and 2 for the robust g-prior). Finally GMJMCMC with Jeffrey's prior reported two trees of size 4 and size 5, respectively, each of which included the wrongly detected leaf X_{43} , whereas GMJMCMC with the robust g-prior reported five trees which included wrong leaves.

For the most complex Scenario 6 once again Jeffrey's prior and the robust g-prior perform quite similar. For the first 6 data generating trees both priors have very large power. For L_7 the power is much lower

and both priors report a large number of subtrees which are counted as false positives. For L_8 the alternative representation of the logic expression has been discussed in the main manuscript. Only 58 false positive trees for Jeffrey's prior and 38 for the robust g-prior combined leaves from different true trees. The number of trees including one wrongly detected leave was 3 and 7, respectively, which once more illustrates that GMJMCMC is very good at controlling the type I error when it comes to including leaves which have nothing to do with the data generating model.

Table B.2. Detailed results for the three simulation scenarios for linear regression. A detailed description of the different classes of false positives ($v(L_j), v(M), \text{WL}(1)$) is given in Section B.1. The columns Jef. and Rob. g correspond to GMJMCMC with Jeffrey's prior and with the Robust g-prior, respectively. In Scenario 4 there was only one false positive detection which is listed explicitly. In Scenario 6 frequencies of the three trees which in combination give L_8 are also listed explicitly.

Scenario 4	Jef.	Rob.g	Scenario 6	Jef.	Rob.g
L_1	100	100	L_1	95	99
L_2	99	100	L_2	98	99
L_3	97	98	L_3	98	99
$L_2 \vee L_3$	1		L_4	96	95
Scenario 5			L_5	100	100
L_1	100	100	L_6	95	96
L_2	100	99	L_7	32	45
L_3	96	100	L_8	21	16
L_4	89	90	$X_{11} \wedge X_{13}$	76	78
$v(L_2)$	0	2	$X_{19} \wedge X_{50}$	75	81
$v(L_3)$	12	0	$X_{11} \wedge X_{13} \wedge X_{19} \wedge X_{50}$	72	69
$v(L_4)$	22	19	$v(L_3)$	6	2
$v(M)$	1	2	$v(L_4)$	12	15
$\text{WL}(1)$	2	5	$v(L_5)$	0	2
			$v(L_6)$	18	13
			$v(L_7)$	84	70
			$v(L_8)$	24	49
			$v(M)$	58	38
			$\text{WL}(1)$	3	7

C Real Data Analysis on *Drosophila*

The results for the other four traits (`adjpc1`, `area`, `areat` and `tibia`) which were neither analyzed by ? nor by ? are provided in the following two tables. In case of *Drosophila Simulans* we detected three

Table C.1. Posterior probabilities for additive and epistatic effects detected with GMJMCMC for four additional traits: *Drosophila simulans*.

Population	Phenotype	Chr	Marker name	$\tilde{P}(L Y) > 0.5$
Simulans	adjpc1	3	rox	1.000
Simulans	adjpc1	3	dbi	1.000
Simulans	adjpc1	2	gpdh	1.000
Simulans	adjpc1	X	v	1.000
Simulans	adjpc1	2	plu	1.000
Simulans	adjpc1	3	mst	0.999
Simulans	adjpc1	2^3	(gl)^(fz)	0.998
Simulans	adjpc1	3	efi	0.985
Simulans	adjpc1	X	w	0.984
Simulans	adjpc1	3	fz	0.983
Simulans	adjpc1	3^3	(lsp)^(ht)	0.982
Simulans	adjpc1	2^3	(duc)^(fas)	0.978
Simulans	area	3	fz	1.000
Simulans	area	2	mhc	1.000
Simulans	area	3	jan	1.000
Simulans	area	X	w	1.000
Simulans	area	3	dbi	1.000
Simulans	area	X	v	0.999
Simulans	area	3	rox	0.998
Simulans	area	3	ninaE	0.996
Simulans	area	3	ve	0.990
Simulans	area	2	ninaC	0.970
Simulans	area	2	zip	0.952
Simulans	area	3	ht	0.864
Simulans	area	2	cg	0.806
Simulans	areat	3	jan	1.000
Simulans	areat	2	mhc	1.000
Simulans	areat	X	w	1.000
Simulans	areat	3	tub	1.000
Simulans	areat	3	rox	1.000
Simulans	areat	3	ninaE	1.000
Simulans	areat	3	fz	1.000
Simulans	areat	X	v	1.000
Simulans	areat	3	dbi	1.000
Simulans	areat	2	ninaC	1.000
Simulans	areat	2	zip	1.000
Simulans	areat	3	ve	1.000
Simulans	areat	3	ht	0.952
Simulans	areat	2	cg	0.925
Simulans	tibia	X	run	0.747

two-way interactions for *adjpc1*. For *Drosophila Mauritiana* further two-way interactions were found; two for *adjpc1*, three for *area*, and two more for *areat*. We did not find higher order interactions for any of these traits.

Table C.2. Posterior probabilities for additive and epistatic effects detected with GMJMCMC for four additional traits: *Drosophila Mauritiana*.

Population	Phenotype	Chr	Marker name	$\tilde{P}(L Y) > 0.5$
Mauritiana	adjpc1	2	cg	1.000
Mauritiana	adjpc1	3	ant	1.000
Mauritiana	adjpc1	3	jan	1.000
Mauritiana	adjpc1	3	acr	1.000
Mauritiana	adjpc1	3	eip	1.000
Mauritiana	adjpc1	3V3	(cyc) \vee (hb)	1.000
Mauritiana	adjpc1	2	gl	1.000
Mauritiana	adjpc1	2	sli	1.000
Mauritiana	adjpc1	X	ewg	1.000
Mauritiana	adjpc1	3VX	(mst) \vee (v)	0.999
Mauritiana	area	3	ant	1.000
Mauritiana	area	3	jan	1.000
Mauritiana	area	2	cg	1.000
Mauritiana	area	2	zip	0.999
Mauritiana	area	3	acr	0.990
Mauritiana	area	3	fz	0.985
Mauritiana	area	3VX	(ve) \vee (w)	0.984
Mauritiana	area	X	ewg	0.958
Mauritiana	area	3VX	(tub) \vee (v)	0.890
Mauritiana	area	3	rox	0.873
Mauritiana	area	3V3	(cyc) \vee (tub)	0.862
Mauritiana	area	2	sli	0.714
Mauritiana	area	2	ninaC	0.613
Mauritiana	area	2	mhc	0.535
Mauritiana	areat	3	ant	1.000
Mauritiana	areat	3	efi	1.000
Mauritiana	areat	2	zip	1.000
Mauritiana	areat	2	cg	1.000
Mauritiana	areat	X	ewg	1.000
Mauritiana	areat	3	rox	1.000
Mauritiana	areat	XV3	(v) \vee (mst)	1.000
Mauritiana	areat	3	fz	0.996
Mauritiana	areat	3^2	(1-(fz)) \wedge (ninaC)	0.974
Mauritiana	areat	3	acr	0.973
Mauritiana	areat	3	cyc	0.685
Mauritiana	tibia	X	v	0.999
Mauritiana	tibia	3	hb	0.999
Mauritiana	tibia	2	mhc	0.997
Mauritiana	tibia	2	plu	0.625