*Journal of Statistical Software*

*MMMMMM YYYY, Volume VV, Issue II.* *doi: 10.18637/jss.v000.i00*

# varbvs: A Software Toolkit for Fast Variable Selection in Genome-wide Association Studies and Other Large-scale Regression Applications

**Peter Carbonetto**
University of Chicago

**Xiang Zhou**
University of Michigan

**Matthew Stephens**
University of Chicago

**Abstract**

We introduce **varbvs**, a suite of functions written in R and MATLAB for analysis of large-scale data sets using Bayesian variable selection methods. The development of **varbvs** was motivated by the recent success of Bayesian variable selection methods for solving important problems in genome-wide association studies (GWAS). Although these methods have several important benefits over comparable penalized regression approaches (e.g., the Elastic Net), they have not been widely adopted because they incur a high computational cost and can be difficult to use. To facilitate their application to large-scale regression problems in GWAS, computational biology and other areas, we have developed numerical optimization algorithms based on variational approximation methods that make it feasible to apply Bayesian variable selection to very large data sets. We have packaged the algorithms into a convenient interface that hides most of the complexities of modeling and optimization, while still providing many options for adaptation to a range of applications. We demonstrate that **varbvs** scales well to large data sets, and has features that facilitate rapid data analyses. Moreover, **varbvs** allows for extensive model customization, which can be used to, for example, incorporate external information into the analysis, and we show how this feature can be used to augment the biological insights generated from genetic data sets. Although our examples mainly focus on GWAS, we expect that the combination of an easy-to-use interface and robust, scalable algorithms for posterior computation will promote the use of Bayesian variable selection in many areas of applied statistics and computational biology. The most recent R and MATLAB source code is available for download at Github (`http://github.com/pcarbo/varbvs`) under the GNU General Public License, and the R package is available through CRAN (`http://cran.r-project.org/package=varbvs`).

*Keywords*: Bayesian variable selection, linear regression, logistic regression, approximate posterior computation, variational inference, Bayes factors, genome-wide association studies, quantitative trait locus mapping, R, MATLAB.

# 1. Introduction

We present a software toolkit for fitting variable selection models to large-scale data sets. We call our software **varbvs**—short for "variational Bayesian variable selection"—as it builds on Bayesian models for variable selection in regression (George and McCulloch 1993; Mitchell and Beauchamp 1988; O'Hara and Sillanpäa 2009) and variational approximation techniques (Blei *et al.* 2016; Jordan *et al.* 1999; Logsdon *et al.* 2010; Ormerod and Wand 2010; Wainwright and Jordan 2008). We have developed efficient implementations for both R (R Core Team 2016) and MATLAB (The MathWorks, Inc. 2016), which we have applied to data sets containing hundreds of thousands of variables and thousands of samples. While our primary motivation is to encourage use of multi-marker regresion models for genome-wide association studies (GWAS) (Carbonetto and Stephens 2012; Guan and Stephens 2011), Bayesian variable selection methods are very general and widely applicable, and we expect that **varbvs** will be useful in many other areas of applied statistics and computational biology.

Bayesian variable selection (BVS) models, and extensions to these models, have recently been shown to provide attractive solutions to a number of important problems in GWAS. This includes mapping of complex disease and trait loci (Carbonetto and Stephens 2012; Guan and Stephens 2011; Hoggart *et al.* 2008; Logsdon *et al.* 2010), enrichment analysis of genetic associations (Carbonetto and Stephens 2013), estimating the proportion of variance in phenotypes explained by available genotypes (Guan and Stephens 2011; Zhou *et al.* 2013), trait and breeding value prediction (Lee *et al.* 2008; Meuwissen *et al.* 2001; Moser *et al.* 2015; Perez and de los Campos 2014; Zhou *et al.* 2013), and disentangling the association signal across all genetic variants within a particular candidate locus, otherwise known as "fine-mapping" (Wallace *et al.* 2015).[1] Despite these contributions, BVS methods have not been widely adopted for GWAS. This is partly because they are difficult to use; barriers include appropriate specification of priors and efficient computation of posterior probabilities. Therefore, our first aim in developing this software is to make these methods accessible to practitioners who may not be familiar with the benefits of BVS, and who are already familiar with standard methods for analysis of GWAS, such as PLINK (Purcell *et al.* 2007) and linear mixed models (Kang *et al.* 2010; Listgarten *et al.* 2012; Yang *et al.* 2011; Zhou and Stephens 2012). In our examples, we demonstrate how **varbvs** has unique features that make it attractive for tackling large-scale variable selection problems in GWAS.

Our second aim is to provide an alternative to commonly used toolkits for penalized sparse regression.[2] Our software is perhaps most comparable to the popular R package **glmnet** (Friedman *et al.* 2010), which combines penalized sparse regression techniques—specifically, the Lasso (Tibshirani 1994) and the Elastic net (Zou and Hastie 2005)—with advanced optimization techniques (Friedman *et al.* 2007) for an efficient and flexible approach to variable selection. In fact, we explicitly designed the **varbvs** interface to be similar to **glmnet** so that researchers already familiar with penalized sparse regression techniques can easily explore the benefits of the BVS approach. In our first example (Sec. 2), we illustrate the shared features

[1] *Peter:* I'd like to get your feedback on the first two sentences of the paragraph below. Did I fail to mention any important applications of BVS methods in GWAS? Any other important papers I should cite here? *Xiang:* I think you have included all the important areas.

[2] *Peter:* "Penalized sparse regression" is the term I'm using in this paper to broadly refer to regression methods such as Lasso and Elastic Net that use penalty terms to "shrink" the coefficients. I don't know if there is a more widely accepted term for this; other suggestions are welcome. Xiang suggested "penalized regression models," but that seems to broad to me.

and differences of **glmnet** and **varbvs**.

Both BVS and penalized sparse regression methods such as the Elastic Net are guided by a common principle: fit a regression model for a set of variables, selecting only the variables that are useful for explaining the outcome. However, an important advantage of BVS is that it provides a measure of uncertainty in the parameter estimates. For example, **varbvs** computes, for each candidate variable, the probability that the variable is included in the regression model—what we call here the "posterior inclusion probability" (PIP). The PIP can be directly interpreted as a measure of support for each candidate variable, which is calibrated so long as that the priors are appropriately specified; no subsequent cross-validation or assessment of false positive rates are required to determine significance levels.[3] A second important advantage of BVS over penalized sparse regression is that it allows for the possibility of model comparison through approximate computation of Bayes factors (Kass and Raftery 1995). In one of the examples below (Sec. 5), we illustrate how model comparison in BVS can be used to quantify support for candidate gene sets contributing to disease susceptibility.

Although there are clear benefits to BVS for GWAS and other large-scale regression applications, two factors have prevented its more widespread use. The first limitation is that computing exact posterior probabilities, which reduces to a high-dimensional integration problem, is intractable except in very small data sets, and standard approaches for approximating these high-dimensional integrals using Monte Carlo techniques scale poorly to large data sets (Bottolo and Richardson 2010; Clyde *et al.* 2011; Dellaportas *et al.* 2002; Erbe *et al.* 2012; Guan and Stephens 2011; Perez and de los Campos 2014; Wallace *et al.* 2015; Zhou *et al.* 2013).[4] A second barrier to more widespread use is that the choice of priors requires considerable expertise in Bayesian data analysis. Our software **varbvs** remedies these two limitations by implementing fast posterior computation using variational approximation techniques, and by providing default priors that are suitable for many problem areas, while also allowing for extensive prior customization.

Before describing the BVS method in detail, we begin our presentation of **varbvs** by walking through an extended example to give some intuition for the functionality of **varbvs**, and how it compares to **glmnet**, a regression analysis framework that will be familiar to many readers. This extended example is presented in Sec. 2.

After working through this introductory example, Sec. 3 briefly reviews Bayesian variable selection in regression, and specifically how BVS is implemented in the **varbvs** R package. The function `varbvs`, in particular, provides a high-level yet highly customizable gateway to the BVS model fitting procedures. The algorithm for fast posterior computation using variational approximation techniques is also briefly described. In subsequent examples, we show how this approximation should be accounted for when interpreting the results of a **varbvs** analysis.

Sections 4 and 5 give more advanced examples that illustrate the use of **varbvs** in large data sets with tens or hundreds of thousands of variables. The first example, in particular, shows how a **varbvs** analysis of genotypes (the candidate variables) and physiological trait measurements (the regression outcome) can suggest connections between genes and traits (the trait in this example is testis weight measured in mice). The second example illustrates a similar **var-**

[3] *Peter:* Not sure if it is a good idea to mention the cross-validation or estimation of false positive rates.

[4] *Peter:* Are there any other papers we should be citing here on Monte Carlo/MCMC for BVS? (I might add a citation to a paper by Arnaud Doucet.) *Xiang:* this list looks good to me.

**bvs** analysis for mapping genes contributing to Crohn's disease risk, a chronic inflammatory health condition. In both examples, although the data sets are large and complex, **varbvs** can be rapidly applied to generate results that are interpretable and, in these examples, biologically meaningful. We round out the presentation of the **varbvs** software with additional discussion on interpreting the results of a **varbvs** analysis, background on some of the numerical aspects of the implementation, and a few other usage recommendations that did not fit well with any of the worked examples.

This paper focuses on the R package (Carbonetto and Stephens 2016) since R has emerged as the dominant programming language in several areas of research, including applied statistics and computational biology. Nonetheless, we encourage investigators with access to MATLAB to consider the MATLAB interface—it has all the features of the R implementation and, since it builds on MATLAB's state-of-the-art numerical computing platform, it can be substantially faster for large data sets. Therefore, we have preferred using the MATLAB implementation for large-scale applications such as the GWAS examples presented in Sections 4 and 5.

## 2. Example illustrating features of glmnet and varbvs

We illustrate **glmnet** and **varbvs** on a smaller data set that has been used in several previous articles to compare methods for penalized regression (e.g., Breheny and Huang 2011; Friedman *et al.* 2010; Tibshirani *et al.* 2005; Zou and Hastie 2005). This example is meant to demonstrate the **varbvs** R interface, and provide some intuition for the different properties of BVS and penalized sparse regression as implemented by **varbvs** and **glmnet**, respectively. It is not intended to be a comprehensive treatment of these topics. R script `demo.leukemia.R` included as part of the **varbvs** R package reproduces the results of this analysis, including all the plots shown in Figures 1 and 2.

The data for this example consist of expression levels recorded for 3,571 genes in 72 patients with leukemia (Golub *et al.* 1999). The 3,571 genes are the candidate variables. The binary outcome, modeled using a logistic regression, encodes the disease subtype: acute lymphobastic leukemia (ALL) or acute myeloid leukemia (AML). We use the preprocessed data of Dettling (2004) which we retrieved from the supplementary materials accompanying Friedman *et al.* (2010). These data are represented as a 72 × 3571 matrix `X` of gene expression levels, and a vector `y` of 72 binary disease outcomes. We fit logistic models to these data using **glmnet** and **varbvs**, and explore properties of the fitted models.

We begin with the **glmnet** analysis of the leukemia data. For each setting of the penalty strength parameter, **glmnet** fits a logistic regression model to the data by solving the following convex optimization problem:

$$\underset{\beta_0 \in \mathbb{R}, \beta \in \mathbb{R}^p}{\text{minimize}} \quad -\frac{1}{n}\sum_{i=1}^{n} \Pr(y_i \,|\, x_i, \beta_0, \beta) + \frac{\lambda}{2}(1-\alpha)\|\beta\|_2^2 + \lambda\alpha\|\beta\|_1, \tag{1}$$

where $x_i$ is the vector of expression levels recorded in patient $i$, $y_i$ is the disease outcome, $n = 72$ is the number of samples, $p = 3571$ is the number of candidate variables, $\beta$ is the vector of logistic regression coefficients, $\beta_0$ is the intercept, $\|\cdot\|_1$ is the $\ell_1$-norm, $\|\cdot\|_2$ is the Euclidean ($\ell_2$) norm, $\Pr(y_i \,|\, x_i, \beta_0, \beta)$ is the logistic regression likelihood (see Equation 4 below). Following Friedman *et al.* (2010), $\lambda$ determines the overall penalty strength, and $\alpha$ balances the $\ell_1$ and $\ell_2$ penalty terms (here, we set $\alpha = 0.95$). This model fitting is easily

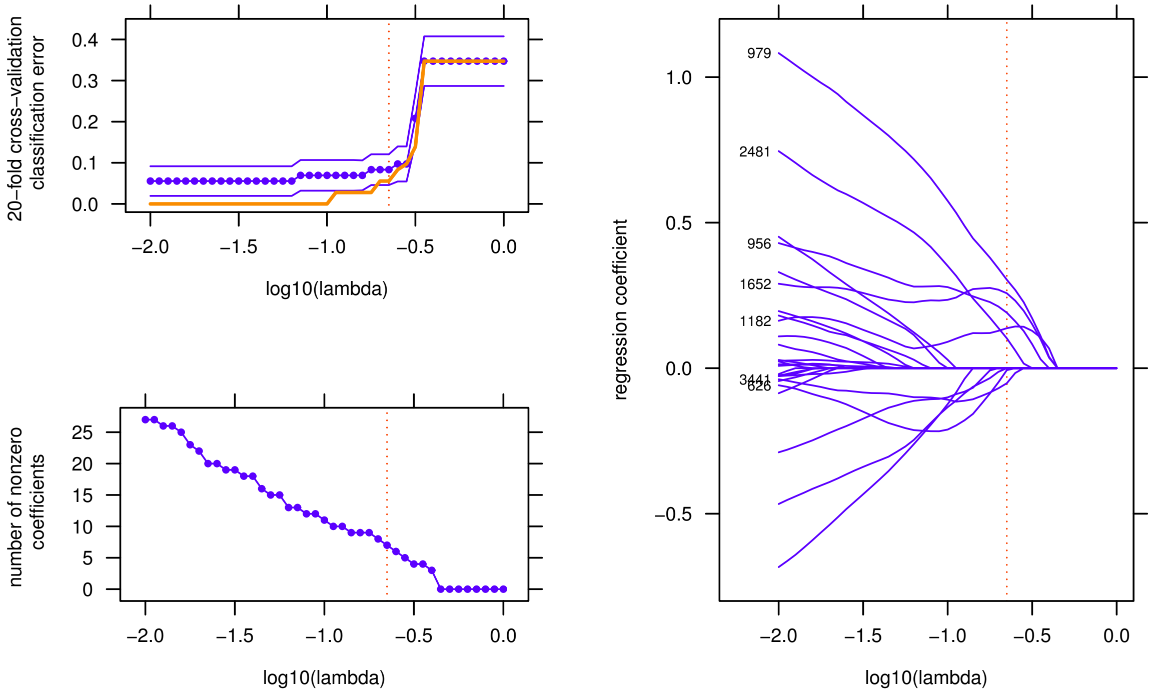

Figure 1: **glmnet** analysis of leukemia data. *top-left panel:* $\ell_1$-penalty strength parameter (`lambda`) against proportion of left-out samples in 20-fold cross-validation that are misclassified by fitted Elastic Net model. Top and bottom curves give the confidence intervals for the classification error across the 20 folds; middle curve in blue is the mean classification error. The wider orange line gives the classification error for the model fitted to the entire data set (see the call to function `glmnet` in the text). *bottom-left panel:* Number of variables included in fitted model—*i.e.*, number of non-zero coefficients—at each setting of `lambda`. *right-hand panel:* Regression coefficients of individual variables at different settings of `lambda`. Labeled curves highlight the 7 variables included in the logistic regression model at the setting of `lambda` selected by 20-fold cross-validation (`lambda = 0.224`, shown in the plots as a dashed vertical red line). These plots were generated by script `demo.leukemia.R`.

accomplished with a single call to the `glmnet` function:

```
R> data(leukemia)
R> X <- leukemia$x
R> y <- leukemia$y
R> fit.glmnet <- glmnet(X, y, family = "binomial", alpha = 0.95,
+                       lambda = 10^(seq(-2,0,0.05)))
```

(We overrode the default `lambda` to make the plots below easier to follow, but it yields a similar result to the default setting.) The regression coefficients estimated for each model, `coef(fit.glmnet)`, are represented as a 3571 × 41 matrix, with one column for each setting of `lambda`, and one row for each variable (including the intercept).

The right-hand plot in Fig. 1 shows the characteristic shrinkage pattern of sparse regression methods such as the Elastic Net; as $\lambda$ (`lambda`) becomes larger, the $\ell_1$-penalty term becomes more prominent, thereby encouraging more shrinkage of the regression coefficients. The bottom-left plot shows the total number of variables (*i.e.*, gene expression levels) with non-zero coefficients at each setting of `lambda`. Observe that, at settings of `lambda` greater than 0.5, all variables in the logistic regression model other than the intercept have coefficients of zero, and therefore none of the gene expression levels help predict the leukemia outcome for all `lambda > 0.5`.

The top-left plot in Fig. 1 shows the evolution of the cross-validation classification error at the same candidate settings of `lambda`. Small values of `lambda` allow for more complex models, and therefore offer a better fit to the data. However, there is a concern that complex models may "overfit" to the data—that is, they are able to accurately predict the outcome in the data used to fit the model, but this accuracy does not extend to unseen (test) examples. Cross-validation is a simple way to guard against overfitting by selecting a penalty strength that provides high accuracy in samples that are not used to fit the model. **glmnet** provides a single function, `cv.glmnet`, to implement the cross-validation procedure:

```
R> out.cv.glmnet <-
+   cv.glmnet(X, y, family = "binomial", type.measure = "class",
+             lambda = 10^(seq(-2,0,0.05)), alpha = 0.95, nfolds = 20)
R> print(out.cv.glmnet$lambda.1se)

[1] 0.2239
```

The penalty strength selected by 20-fold cross-validation, `lambda.1se`, is shown as a dashed vertical line in each of the plots in Fig. 1. It is the largest value of `lambda` that yields a classification error within 1 standard error of the minimum classification error. At this $\ell_1$-penalization level, **glmnet** yields an extremely sparse regression model—only 7 out of the 3,571 gene expression features are included in the model (Fig. 1, right-hand panel)—yet these 7 features are sufficient to correctly predict the leukemia outcome in 68 out of the 72 examples:

```
R> y.glmnet <-
     c(predict(fit.glmnet, X, s = out.cv.glmnet$lambda.1se, type = "class"))
R> print(table(true = factor(y), pred = factor(y.glmnet)))

    pred
true  0  1
   0 47  0
   1  4 21
```

The entire **glmnet** analysis, including cross-validation, took less than 3 seconds to run on a computer with a 1.86 GHz Intel Core 2 Duo processor.

Next, we compare the **glmnet** analysis of the leukemia data against an analysis of the same data using **varbvs**. As before, we use logistic regression (Equation 4) to model the outcome given the regression coefficients. But rather than optimize the coefficients subject to a penalty, we introduce an exchangeable prior on the coefficients,

$$\Pr(\beta_i \,|\, \pi, \sigma_a^2) = \begin{cases} \pi N(0, \sigma_a^2) & \text{if } \beta_i \neq 0 \\ 1 - \pi & \text{otherwise.} \end{cases} \tag{2}$$

and we compute approximate posterior probabilities with respect to this prior. Instead of a two-step analysis—modeling fitting and cross-validation—the **varbvs** analysis is accomplished in a single function call:

```
R> fit.varbvs <- varbvs(X, NULL, y, family = "binomial", sa = 1,
+                        logodds = seq(-3.5,-1.5,0.1))
```

(Note that function `varbvs` allows for additional covariates `Z` included in the regression model with probability 1. Sine this option is not used here, we set `Z = NULL`.) This command has a slightly higher runtime than the **glmnet** analysis, taking a little less than 30 seconds to complete on the leukemia data set. The most expensive step is computation of the posterior distribution by fitting the variational approximation.

The complexity of the regression model is controlled by the prior, which is determined by two parameters: the prior log-odds $\log_{10}\left(\frac{\pi}{1-\pi}\right)$ that a variable is included in the regression model (`logodds`), and $\sigma_a^2$, the prior variance of the regression coefficients (`sa`). We compute results for different settings of `logodds`, and keep `sa` constant to simplify the example. (Below, we give guidance on setting this parameter by hand, or fitting this parameter to the data.) Also note that the default setting of `logodds` yields similar results to the ones presented here, and we have supplied a custom setting only to make the example easier to follow.

To illustrate the effect that hyperparameter `logodds` has on model complexity, we compute the classfication error at each setting of the prior log-odds:

```
R> m   <- length(logodds)
R> err <- rep(0,m)
R> for (i in 1:m) {
+  r      <- logodds[i]
+  ypred  <- predict(subset(fit.varbvs, logodds == r), X)
+  err[i] <- mean(y != ypred)
+ }
```

These results are summarized in the top-left plot in Fig. 2. Like **glmnet**, we observe that the **varbvs** model predictions improve as the variable selection prior allows for more complex models. However, in contrast to **glmnet**, the shrinkage is not smooth, nor applied as uniformly; one variable has a much larger posterior mean regression coefficient for all candidate settings of the prior log-odds, and the coefficient remains consistently large at different prior settings (Fig. 2, top-right and bottom-right plots).

A second key difference is that cross-validation is not needed in **varbvs** to select an appropriate level of regularization; as a consequence of taking the Bayesian inference approach, **varbvs** automatically weighs the accuracy of the model predictions against the complexity of the model (Jefferys and Berger 1992; MacKay 1992). In this example, the more complex models (at larger prior log-odds values) offer only a marginally better fit to the data. Therefore, the posterior distribution is most concentrated on less complex models, and on smaller values of the prior log-odds (Fig. 2, bottom-left).

The third main difference is that **varbvs** can account for uncertainty in the hyperparameters by averaging over the candidate settings, weighted by the estimated posterior probabilities (Fig. 2). Model averaging in BVS is typically computationally prohibitive in large-scale

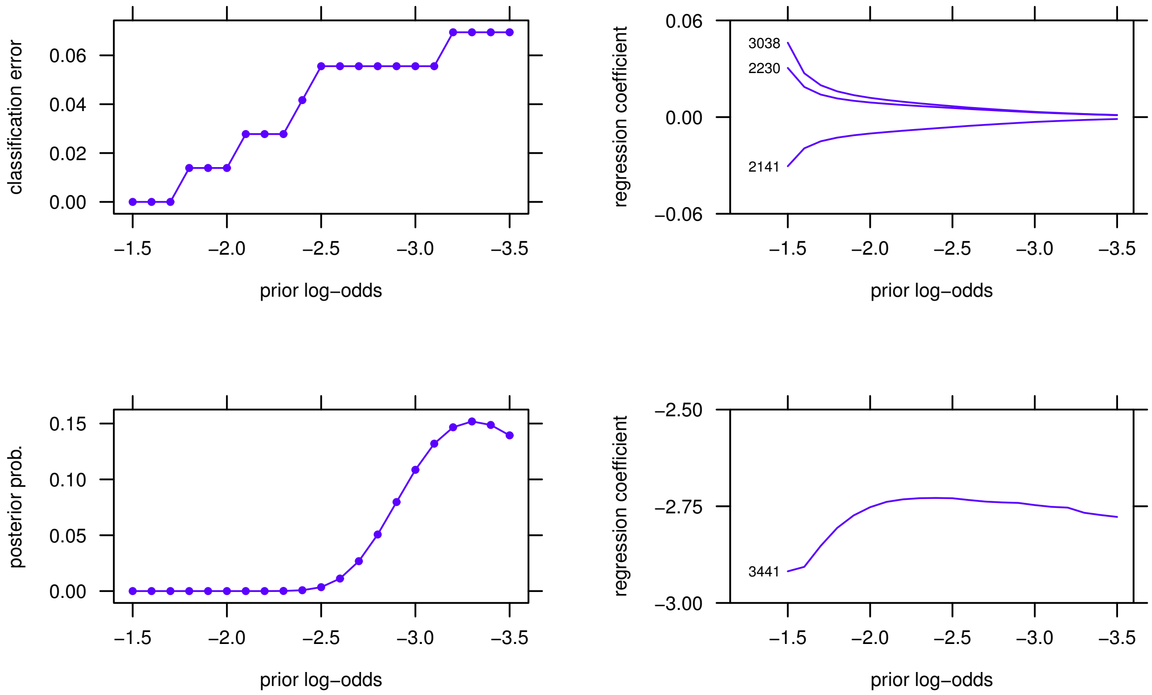

Figure 2: **varbvs** analysis of leukemia data. *top-left panel:* Prior log-odds parameter (`logodds`) against proportion of samples that are misclassified by fitted **varbvs** model. This should be compared against the wider orange line in the top-left panel of Fig. 1. *bottom-left panel:* Posterior probability for each candidate setting of the prior log-odds parameter. *right-hand panels:* For each prior log-odds setting, posterior mean estimates of regression coefficients for the top 4 variables by their (averaged) posterior inclusion probability. The posterior mean coefficient for variable 3441 is much larger than all the others, so it is plotted separately. Note that the prior log-odds in each of the plots is shown in decreasing order from left to right to make the plots more directly comparable to those in Fig. 1.

data sets, but the “fully-factorized” variational approximations proposed in previous papers (Attias 1999; Carbonetto and Stephens 2012; Logsdon *et al.* 2010) yield a simple and efficient approach to account for uncertainty in the hyperparameters. As a result, model averaging is straightforward and practical in **varbvs** even for large data sets, so averaged estimates are computed by default when invoking the BVS analysis routines in **varbvs**. (When model averaging is not practical, an alternative is to fit one or more of the hyperparameters to the data, and this option is provided in **varbvs** as well.) For example, function `predict` automatically estimates the regression outcome in each sample as an average of the estimates weighted by the posterior probability of each hyperparameter setting:

```
R> y.varbvs <- predict(fit.varbvs, X)
R> print(table(true = factor(y), pred = factor(y.varbvs)))
```

```
    pred
```

```
true  0  1
   0 45  2
   1  3 22
```

Observe that these averaged predictions introduce only one more error than the **glmnet** model, and this is achieved by concentrating the posterior distribution on much simpler models, in which most of the variance in the leukemia outcome is explained by a single predictor (variable 3441). In the examples below, we demonstrate this feature in very large data sets.

# 3. Bayesian variable selection, and the varbvs R interface

Here, we define the general analysis setup: the regression model (Sec. 3.1), the variable selection priors (Sec. 3.2), and the approach taken to efficiently estimate the posterior quantities of interest (Sections 3.3 and Sec. 3.4). As we walk through the setup, we connect the elements of the analysis to the R interface. At the end, we summarize the R interface for reference (Sec. 3.5).

Bayesian approaches to variable selection and subset selection have a long history, so we do not attempt to provide a comprehensive overview in this short section. For background on this topic, refer to George (2000) and O'Hara and Sillanpäa (2009).

## 3.1. Regression model

To begin, we model the variable of interest, $Y$, as a function of the candidate predictors $X = (X_1, \ldots, X_p)^T$ and additional covariates $Z = (Z_1, \ldots, Z_m)^T$. We assume that at least one covariate $Z_1$, the intercept, is always included in the model. We consider two possible regression models. When $Y$ is continuously valued, we assume a basic linear regression, in which $Y$ is modeled as a linear combination of the candidate predictors and covariates, plus residual noise $\epsilon \sim N(0, \sigma^2)$:

$$Y = \sum_{i=1}^{m} Z_i u_i + \sum_{i=1}^{p} X_i \beta_i + \epsilon. \tag{3}$$

Alternatively, if the outcome $Y$ is binary-valued (e.g., in a case-control study), we consider an additive model for the log-odds of $Y = 1$:

$$\log\left\{\frac{\Pr(Y=1)}{\Pr(Y=0)}\right\} = \sum_{i=1}^{m} Z_i u_i + \sum_{i=1}^{p} X_i \beta_i. \tag{4}$$

(Since parameter $\sigma^2$ is not needed for the logistic regression, in the definitions below we set $\sigma^2 = 1$ for this case.) Identically to **glmnet**, the linear regression (Equation 3) and logistic regression (Equation 4) models are chosen by setting `family = "gaussian"` and `family = "binomial"`, respectively, in the call to function `varbvs`.

In the general setup, the data consist of an $n \times p$ matrix $\mathbf{X}$ containing observations $x_{ij}$ of the candidate variables, an $n \times m$ matrix $\mathbf{Z}$ containing measurements $z_{ij}$ of the covariates, and a vector $y = (y_1, \ldots, y_n)^T$ containing observations of the regression outcome. These data are provided to function `varbvs` through input arguments `X`, `Z` and `y`.

### 3.2. Variable selection prior

We take the most popular Bayesian approach to variable selection, based on the "spike-and-slab" prior (Mitchell and Beauchamp 1988): with probability $\pi$, coefficient $\beta_i$ is drawn from the "slab," which we take to be a normal density with zero mean and variance $\sigma^2\sigma_a^2$; and with probability $1-\pi$, $\beta_i$ equals zero (the "spike"). The effect of $\pi$ on model sparsity was illustrated in the leukemia example in Sec. 2. Small values of $\pi$, for example, encourage "sparse" regression models, in which only a small proportion of the candidate variables $X_i$ help predict the outcome $Y$. By adopting the spike-and-slab prior, we have framed the variable selection problem—the problem of deciding which variables are useful for predicting outcome $Y$—as the problem of determining which of the regression coefficients $\beta \equiv (\beta_1, \ldots, \beta_p)^T$ are equal to zero.

The rationale for the specific form of the prior we use here has been given in previous papers (Carbonetto and Stephens 2012; Guan and Stephens 2011; Servin and Stephens 2007; Zhou *et al.* 2013), and we do not repeat this discussion here. For example, scaling the prior variance of the coefficients by $\sigma^2$ is necessary to ensure that this prior is invariant to measurement scale (e.g., switching from grams to kilograms). There are of course other prior choices that are often preferrable in some circumstances. For example, although assigning an identical prior of zero mean and variance $\sigma_a^2$ to all the non-zero coefficients is a standard default choice George and McCulloch (1993), an alternative prior that is often used is the $g$-prior (Liang *et al.* 2008; Zellner 1986). For a more detailed discussion on the interpretation of the $g$-prior and other priors specifically in the context of GWAS, see Guan and Stephens (2011) and Servin and Stephens (2007).

Candidate settings of the hyperparameters are specified by three inputs to `varbvs`: `sa`, the prior variance of the regression coefficients; `sigma`, the residual variance (for linear regression only); and `logodds`, the prior log-odds of inclusion, which is equal to $\log_{10}\left\{\frac{\pi}{1-\pi}\right\}$. The prior inclusion probability is parameterized in this way because it is often more natural to draw candidate settings uniformly on the log-odds scale (see also Sec. 3.4). The base-10 logarithm rather than the natural logarithm is used for easier interpretation.

This variable selection prior we have described so far treats all candidate variables $X_i$ equally. However, in some settings we may have additional information that suggests the importance of some variables more than others. For example, in GWAS, we may have some prior knowledge about the genes that are most relevant to the target disease, in which case the candidate genetic variants nearby these genes that might regulate or alter the activity of these genes would be favoured over other variants. **varbvs** can encode preferences such as these with a non-exchangeable prior $\boldsymbol{\pi} = (\pi_1, \ldots, \pi_p)$, which is specified by setting input `logodds` to a matrix in which the rows of this matrix correspond to variables and the columns correspond to hyperparameter settings. Below, we illustrate how this non-exchangeable prior can be used to augment biological insights in a GWAS.

An alternative to specifying candidate settings of the hyperparameters, and averaging over these settings weighted by their marginal posterior probability, is to fit one or more of the hyperparameters to the data. This option can be more practical for analyzing large data sets since it reduces the number of times the co-ordinate ascent optimization algorithm (described below) needs to be run. And it is certainly simpler since it doesn't require the user to specify the candidate hyperparameter settings in advance. This option is activated by setting `update.sigma = TRUE` and/or `update.sa = TRUE` in `varbvs`. It is activated by default when `sigma` or `sa` are not provided as inputs to `varbvs`.

Note that we do not provide an option to fit the `logodds` parameter. Although it would be useful to do so in some settings, we have instead focused on allowing for general (e.g., non-exchangeable) prior inclusion probabilities $\pi_i$ in which we do not specify in advance how these prior probabilites are parameterized. Additionally, in the examples considered here, the final variable selection results can be sensitive to the prior log-odds, so it is preferrable not to rely on a single point estimate of this parameter.

More generally, we caution that a point estimate of a hyperparameter can be misleading in cases where the posterior is spread out over a wide range of settings. For example, when few variables are included in the model, there will be little information guiding the choice of $\sigma_a^2$, so $\sigma_a^2$ will be estimated with a high degree of uncertainty. On the other hand, $\sigma^2$ is typically estimated with high confidence, so it usually reasonable to fit $\sigma^2$ instead of averaging over settings of $\sigma^2$.

The $Z_i$'s are additional predictors that are included in the model with probability 1. Instead of the spike-and-slab prior, they are assigned an improper, uniform prior (*i.e.*, a normal prior with large variance). Although improper priors are generally not advisable because they can result in improper posteriors and Bayes factors (O'Brien and Dunson 2004), this choice allows us to easily integrate out these covariates in the linear regression model (Chipman *et al.* 2001), as well as in the logistic regression case after introducing a variational approximation to the nonlinear logistic factors (Jaakkola and Jordan 2000; see also the Appendix). An intercept is always included in the model with an additional covariate $Z_i = 1$ so the user should never provide the intercept manually as part of input `Z`.

## 3.3. Fast posterior computation via variational approximation

While Markov chain Monte Carlo (MCMC) methods have broadened the appeal of Bayesian variable selection approaches (George and McCulloch 1993), they are numerically intensive, and therefore have also limited application of BVS to large data sets. Here we describe an alternative approach based on variational methods (Blei *et al.* 2016; Jordan *et al.* 1999; Ormerod and Wand 2010; Wainwright and Jordan 2008) that yields fast computation of posterior probabilities at the cost of requiring a more careful interpretation due to the approximations made.

The basic idea is to recast the problem of computing posterior probabilities—which is inherently an intractable, high-dimensional integration problem—as an optimization problem. This is achieved by introducing a class of approximating distributions, then optimizing a criterion (the Kullback-Leibler divergence) to find the distribution within this class that best matches the posterior. To make this approach viable for large problems, we enforce a simple conditional independence property (Carbonetto and Stephens 2012; Logsdon *et al.* 2010): conditioned on the hyperparameters $\theta \equiv \{\sigma^2, \sigma_a^2, \boldsymbol{\pi}\}$, each regression coefficient $\beta_i$ is independent of the other regression coefficients *a posteriori*. We then search for a distribution with this conditional independence property that best "fits" the posterior. Note that this is not the traditional usage of "fitting" in which we seek to optimize the model parameters; here, we are still estimating the posterior distribution of the unknowns given the priors and data. For full details, including derivation of the co-ordinate ascent algorithms, consult Carbonetto and Stephens (2012).

This conditional independence assumption was initially motivated from the GWAS setting where the variables are genetic markers. For most markers, this conditional independence assumption is appropriate—most markers are unlinked because they are on separate chromosomes, or they are weakly linked because of recombination. We expect that this assumption will be reasonable in other settings as well. The inclusion of an intercept is an additional necessary condition for the variables being independent *a posteriori*, which is why we require that the regression model always include an intercept (unlike **glmnet**, where it is optional). Even when many of the variables are strongly correlated, this approximation can often yield accurate inferences so long as individual posterior statistics are interpreted with appropriate consideration of the conditional independence assumption. In the examples below, and in the discussion, we give guidance on interpreting the approximate posterior statistics.

The algorithm for fitting the variational approximation consists of an inner loop and an outer loop. The outer loop iterates over the hyperparameter settings, and is the focus of the next section (Sec. 3.4). The inner loop, given a setting of the hyperparameters, cycles through co-ordinate ascent updates to tighten the lower bound on the marginal likelihood, $\Pr(y \,|\, \mathbf{X}, \mathbf{Z}, \theta) = \int \Pr(y \,|\, \mathbf{X}, \mathbf{Z}, \beta, \sigma^2)\; \Pr(\beta \,|\, \theta)\, d\beta$. The inner loop co-ordinate ascent updates terminate when either the maximum number of inner loop iterations is reached, as specified by input argument `maxiter`, or the maximum difference between the estimated posterior inclusion probabilities is less than `tol`. The computational complexity of the co-ordinate ascent updates scales linearly with the number of variables and the number of samples. However, the number of co-ordinate ascent updates required to reach convergence depends on the covariance structure of the candidate variables. Fastest convergence occurs when the variables are uncorrelated or weakly correlated.

Function `varbvs` outputs three posterior quantities for each variable $X_i$ and for each hyperparameter setting $\theta^{(j)}$:

$$\alpha_{ij} \approx \Pr(\beta_i \neq 0 \,|\, \mathbf{X}, \mathbf{Z}, \theta = \theta^{(j)}) \tag{5}$$

$$\mu_{ij} \approx \mathrm{E}[\beta_i \,|\, \mathbf{X}, \mathbf{Z}, \theta = \theta^{(j)}, \beta_i \neq 0] \tag{6}$$

$$s_{ij}^2 \approx \mathrm{Var}[\beta_i \,|\, \mathbf{X}, \mathbf{Z}, \theta = \theta^{(j)}, \beta_i \neq 0]. \tag{7}$$

Each of these outputs is represented as a $p \times n_s$ matrix, where $p$ is the number of variables and $n_s$ is the number of candidate hyperparameter settings. For the $i$th variable and $j$th hyperparameter setting, `alpha[i,j]` is the variational estimate of the PIP (Equation 5), `mu[i,j]` is the variational estimate of the posterior mean coefficient given that it is included in the regression model (Equation 6), and `s[i,j]` is the estimated posterior variance (Equation 7). For example, in the **varbvs** analysis of the leukemia data set, the posterior mean estimates of the regression coefficients for variable 3441, and for settings `logodds = -2.0` to `logodds = -1.5`, shown in the bottom-right panel of Fig. 2, are

```
R> print(tail(fit.varbvs$alpha[3441,] * fit.varbvs$mu[3441,]))
```

```
[1] -2.752 -2.774 -2.806 -2.852 -2.907 -2.918
```

These posterior statistics are also the free parameters of the approximating distribution. Therefore, they are also the parameters that are optimized as part of the "inner loop" co-ordinate ascent updates. An additional set of free parameters is needed for fast computation with the logistic regression model. The fitted value, `eta`, is returned as an $n \times n_s$ matrix. When a good guess of these posterior statistics is available in advance, this can be supplied by inputs `alpha`, `mu`, `s`, and `eta` for `family = "binomial"`, to function `varbvs`.

Note that special care is required for interpreting the logistic regression results due to the additional approximation introduced. As a general guideline, only the relative magnitudes of the coefficients are meaningful.

### 3.4. Averaging over the hyperparameter settings

As we mentioned above, we provide the option of fitting hyperparameters $\sigma^2$ (`sigma`) and $\sigma_a^2$ (`sa`) to the data (see `varbvs` input arguments `update.sigma` and `update.sa`). For fitting these parameters, we implemented an approximate expectation maximization (EM) approach (Heskes *et al.* 2004; Neal *et al.* 1998), in which the E-step is approximated using the variational techniques described above. It is often good practice, however, to account for uncertainty in the hyperparameters when reporting final posterior quantities. For example, hyperparameter $\sigma_a^2$ can be estimated with a high degree of uncertainty when only a few variables are included in the model. Therefore, we also implement the *Bayesian model averaging* strategy (Hoeting *et al.* 1999), in which we average over settings of the hyperparameters, weighted by the posterior probability of each setting. The **varbvs** interface also allows for fitting some hyperparameters and averaging over others, in which case the fitted hyperparameters are fit separately for each candidate setting of the unfitted hyperparameters.

Although it is good practice, model averaging for BVS raises two difficulties. The first difficulty is that averaging requires specification of the prior, $\Pr(\sigma^2, \sigma_a^2, \boldsymbol{\pi})$. **varbvs** allows complete flexibility in this choice. However, this leaves it up to the user to come up with a prior that is well suited to the particular data set. The default is an exchangeable prior (*i.e.*, $\pi_i = \pi$) in which candidate settings of $\pi$ are evenly spaced on the log-odds scale. This implies a default prior for $\pi$ that is uniform on the log-odds scale, which we believe is a natural default for settings with large numbers of candidate variables.

For $\sigma^2$ and $\sigma_a^2$, and the default is to fit these parameters to the data. To override this default, and select a set of candidate settings for `sa`, we have advocated for setting `sa` indirectly through a prior estimate of the proportion of variance in the outcome explained by the variables, since it is often more natural to specify the prior proportion of variance explained rather than the prior variance (Guan and Stephens 2011; Zhou and Stephens 2012). See `help(varbvs)` for more details about this.

The second difficulty is that model averaging is computationally intensive, especially for large data sets. We formulate a simple piecewise numerical approximation to the integral over $\theta$ (Burden and Faires 2005), in which the marginal likelihood $\Pr(y \mid \mathbf{X}, \mathbf{Z}, \theta)$ for each candidate setting $\theta = \theta^{(j)}$ is replaced by its estimated variational lower bound (see Carbonetto and Stephens 2012, and the Appendix). For example, in the **varbvs** analysis of the leukemia data, posterior probabilities are estimated for 21 settings of `logodds` ranging from `-3.5` to `-1.5`, yielding a piecewise approximation to the posterior density, shown here and in the bottom-left panel of Fig. 2:

```
R> w <- normalizelogweights(fit.varbvs$logw)
R> print(sprintf("\%0.2f", w))

[1] "0.14" "0.15" "0.15" "0.15" "0.13" "0.11" "0.08" "0.05" "0.03" "0.01"
[11] "0.00" "0.00" "0.00" "0.00" "0.00" "0.00" "0.00" "0.00" "0.00" "0.00"
[21] "0.00"
```

Although simple, this provides a reasonable, if rough, approximation in 1–3 dimensions, and importantly, provides flexibility for tackling large data sets.

A subtle but nonetheless important issue with this strategy is that the variational lower bound can be sensitive to the choice of starting point $\phi^{(\mathsf{init})}$, and therefore could affect the quality of the approximation. To provide a more accurate variational approximation of the posterior distribution, the optimization procedure has two stages by default. In the first stage, the entire procedure is run to completion, and the variational parameters (`alpha`, `mu`, `s`, `eta`) corresponding to the maximum lower bound are then used to initialize the co-ordinate ascent updates in the second stage. Although this has the effect of doubling the computation time in the worst case, the final posterior estimates tend to be more accurate using this two-stage optimization approach (Carbonetto and Stephens 2012). Set `initialize.params = FALSE` in `varbvs` to skip the initialization phase and reduce computate time at the risk of less accurate posterior estimates.

### 3.5. The `varbvs` function

To recap, we summarize function `varbvs`, the core function of the R package that provides a single interface for all BVS posterior computation and model fitting procedures. To provide a familiar interface, we have modeled it after **glmnet**. We have grouped the inputs to `varbvs` by their function:

```
varbvs(X, Z, y, family,                               # Data.
       sigma, sa, logodds,                            # Hyperparameters.
       alpha, mu, eta,                                # Variational parameters.
       update.sigma, update.sa, optimize.eta,         # Optimization and model
       initialize.params, nr, sa0, n0, tol, maxiter, # fitting settings.
       verbose)                                       # Other settings.
```

The first four input arguments are reserved for the data: the $n \times p$ input matrix `X` and the $n \times m$ input matrix `Z`, where $n$ is the number of data examples, $p$ is the number of candidate variables and $m$ is the number of covariates (not including the intercept); the $n$ observations of the regression outcome stored in vector `y`; and the option to specify a linear regression model (`family = "gaussian"`, the default) or logistic regression when all entries of `y` are 0 or 1 (`family = "binomial"`).

The next three input arguments, `sigma`, `sa` and `logodds`, are optional, and specify the candidate hyperparameter settings. Each of these inputs must have the same number of entries $n_s$, except in the special case when the prior log-odds is specified separately for each variable, in which case `logodds` is a $p \times n_s$ matrix. If inputs `sigma` or `sa` are missing, they are automatically fitted to the data by computing approximate maximum-likelihood or *maximum a posteriori* estimates. See Sec. 3.4 for more details on these inputs.

When good initial estimates of the variational parameters are available, they can be provided to `varbvs` through input arguments `alpha`, `mu` and `s`. Each of these inputs must be an $p \times n_s$ matrix, or a $p \times 1$ matrix when all variational approximations are provided the same initial parameter estimate. Input `eta` is an additional set of free parameters for the variational approximation for the logistic regression model. It is either an $n \times n_s$ matrix or an $n \times 1$ matrix. The remaining input arguments control various aspects of the model fitting and optimization procedures, and are detailed in the help page for function `varbvs`.

The varbvs function returns an S3 object of class 'varbvs'. The main components of interest are:

- logw—Array in which logw[i] is the variational lower bound on the marginal log-likelihood for the $i$th hyperparameter setting.
- alpha—Variational estimates of posterior inclusion probabilities alpha[i,j] for each variable $X_i$ and hyperparameter setting $\theta^{(j)}$.
- mu—Variational estimates of posterior mean coefficients mu[i,j] for each variable $X_i$ and hyperparameter setting $\theta^{(j)}$.
- s—Variational estimates of posterior variances s[i,j] for each variable $X_i$ and hyperparameter setting $\theta^{(j)}$.
- mu.cov—Posterior mean regression coefficients mu.cov[i,j] for each covariate $Z_i$ (including the intercept) for each hyperparameter setting $\theta^{(j)}$.
- eta—Additional variational parameters for family = "binomial" only.
- pve—For each hyperparameter setting $\theta^{(j)}$, and for each variable $X_i$, pve[i,j] is the mean estimate of the proportion of variance in the outcome $Y$ explained by $X_i$, conditioned on $X_i$ being included in the model. This is computed for family = "gaussian" only.
- model.pve—Samples drawn from the posterior distribution giving estimates of the proportion of variance in the outcome $Y$ explained by the fitted variable selection model. For example, to calculate the posterior mean estimate of the proportion of variance explained, enter mean(fit.varbvs$model.pve), where fit.varbvs is the varbvs return value. This is provided for family = "gaussian" only.

The components logw, alpha, mu and s are the basic posterior quantities that can be used to easily calculate many other posterior statistics of interest. For example, the marginal posterior inclusion probabilities for all candidate variables are computed as a simple weighted average:

```
R> w   <- normalizelogweights(fit.varbvs$logw)
R> pip <- c(fit.varbvs$alpha %*% w)
```

To take another example, the probability that at least 1 variable is included is obtained as

```
R> p0 <- apply(1 - fit$alpha, 2, prod)
R> sum(w * (1 - p0)))
```

We provide several supporting functions for the 'varbvs' class, including: function summary for quickly generating analysis summaries; predict for making predictions from a fitted model; and plot for generating a visual summary of the variable selection results. Function predict was already illustrated in the leukemia example. Other functions are demonstrated in the examples below. In our final example, we demonstrate an advanced feature of the **varbvs** package—model comparison—through the combination of the varbvs and bayesfactor functions.

# 4. Example: mapping testis weight loci in outbred mice

In our second example, we illustrate the features of **varbvs** for genome-wide mapping of a complex trait. The data, retrieved from `http://datadryad.org/resource/doi:10.5061/dryad.2rs41`, are body and testis weight measurements recorded for 993 outbred mice, and genotypes at 79,748 single nucleotide polymorphisms (SNPs) for the same mice (Parker *et al.* 2016). The main goal of this analysis is to identify genetic variants contributing to variation in testis weight. In R, the genotype data are represented as a $993 \times 79,748$ matrix `geno`. The phenotype data—body weight (in g) and testis weight (in mg)—are stored in the `sacwt` and `testis` columns of a matrix, `pheno`:

```
R> head(pheno[,c("sacwt", "testis")])
```

```
      sacwt testis
26305  46.6 0.1396
26306  35.7 0.1692
26307  34.1 0.1878
26308  41.8 0.2002
26309  39.5 0.1875
26310  36.0 0.1826
```

Vignette `demo.cfw.R` in the **varbvs** R package reproduces all the results of this analysis, including the plots shown in Fig. 3.

The standard approach to quantitative trait locus (QTL) mapping is to quantify support separately for each SNP. For example, this was the approach taken in Parker *et al.* (2016). Here, we implement this "single-marker" mapping approach using the `-lm 2` option in GEMMA (Zhou and Stephens 2012), which returns a likelihood-ratio test $p$ value for each SNP. We compare this single-marker analysis approach against a **varbvs** "multi-marker" analysis of the same data.[5]

In the **varbvs** analysis, the quantitative trait (testis weight) is modeled as a linear combination of the covariate (body weight) and the candidate variables (the 79,748 SNPs). The analysis of these data is accomplished with a single function call:

```
R> fit <- varbvs(geno, as.matrix(pheno[,"sacwt"]), pheno[,"testis"],
+                 sa = 0.05, logodds = seq(-5,-3,0.25)))
```

This function call completed in less than 4 minutes in R version 3.3.0 on a MacBook Air with a 1.86 GHz Intel CPU and 4 GB of memory. Note that, to simplify this example, we have fixed the prior variance parameter `sa` to `0.05`. This choice is informed by our power calculations, and by our expectation that most of the genetic effects on complex traits are small. In general, it is preferable to average over a range of settings to avoid sensitivity of the QTL mapping results to a specific prior choice (see Sec. 3.4).

Once the model fitting has completed, we quickly generate a results summary by calling the `summary` function:

[5] *Peter:* I ended up removing these two sentences: "One difficulty with implementing the single-marker approach in this case is that we have no immediately available standard for determining significance of the $p$ values because this outbred mouse population has not been previously used for GWAS. Therefore, determining an appropriate significance threshold is an additional step to the single-marker analysis that is not needed in the varbvs analysis."

```
R> print(summary(fit))

Summary of fitted Bayesian variable selection model:
family:     gaussian   num. hyperparameter settings: 9
samples:    993        iid variable selection prior: yes
variables:  79748      fit prior var. of coefs (sa): no
covariates: 2          fit residual var. (sigma):    yes
maximum log-likelihood lower bound: 2428.7093
proportion of variance explained: 0.149 [0.090,0.200]
Hyperparameters:
        estimate Pr>0.95              candidate values
sigma   0.000389 [0.000379,0.000404] NA--NA
sa            NA [NA,NA]              0.05--0.05
logodds    -3.78 [-4.25,-3.50]        (-5.00)--(-3.00)
Selected variables by probability cutoff:
>0.10 >0.25 >0.50 >0.75 >0.90 >0.95
    3     3     3     2     2     1
Top 5 variables by inclusion probability:
  index    variable   prob    PVE     coef  Pr(coef.>0.95)
1 59249   rs6279141 1.0000 0.0631 -0.00806 [-0.010,-0.006]
2 24952  rs33217671 0.9351 0.0220  0.00509 [+0.003,+0.007]
3  9203  rs33199318 0.6869 0.0170  0.00666 [+0.004,+0.010]
4 67415  rs52004293 0.0739 0.0136  0.00347 [+0.002,+0.005]
5 44315 rs253722776 0.0707 0.0133 -0.00369 [-0.005,-0.002]
```

Based on this summary, we find that the fitted model explains 15% of the variance of testis weight with a remarkably small number of variables; only 3 out of the 79,748 SNPs are included in the model with posterior probability greater than 0.5. (To be precise, 15% is the variance explained in testis weight residuals after controlling for body weight.) Further, a single SNP (rs6279141) accounts for over 6% of variance in testis weight. This SNP is located on chromosome 13 approximately 1 Mb from *Inhba*, a gene that has been previously shown to affect testis morphogenesis, testicular cell proliferation and testis weight in mice (Mendis *et al.* 2011; Mithraprabhu *et al.* 2010; Tomaszewski *et al.* 2007).

The results are summarized visually using the customized `plot` function:

```
R> library("lattice")
R> print(plot(fit, vars = c("rs33199318", "rs33217671", "rs6279141"),
+              groups = map$chr, gap = 1500))
```

The output of the call to `plot` is shown in Fig. 3a. The `plot` function takes an extra `"group"` argument, which in this example allows us to arrange the variable selection results by chromosome. The 3 SNPs included in the model with high probability, on chromosomes 2, 5 and 13, clearly stand out in this plot.

It is informative to compare these probabilities against the "single-marker" $p$ values that ignore correlations between SNPs (computed using GEMMA). Plotting these $p$ values on the log-scale (Fig. 3b) produces a "Manhattan plot" that is conventionally used to summarize the results

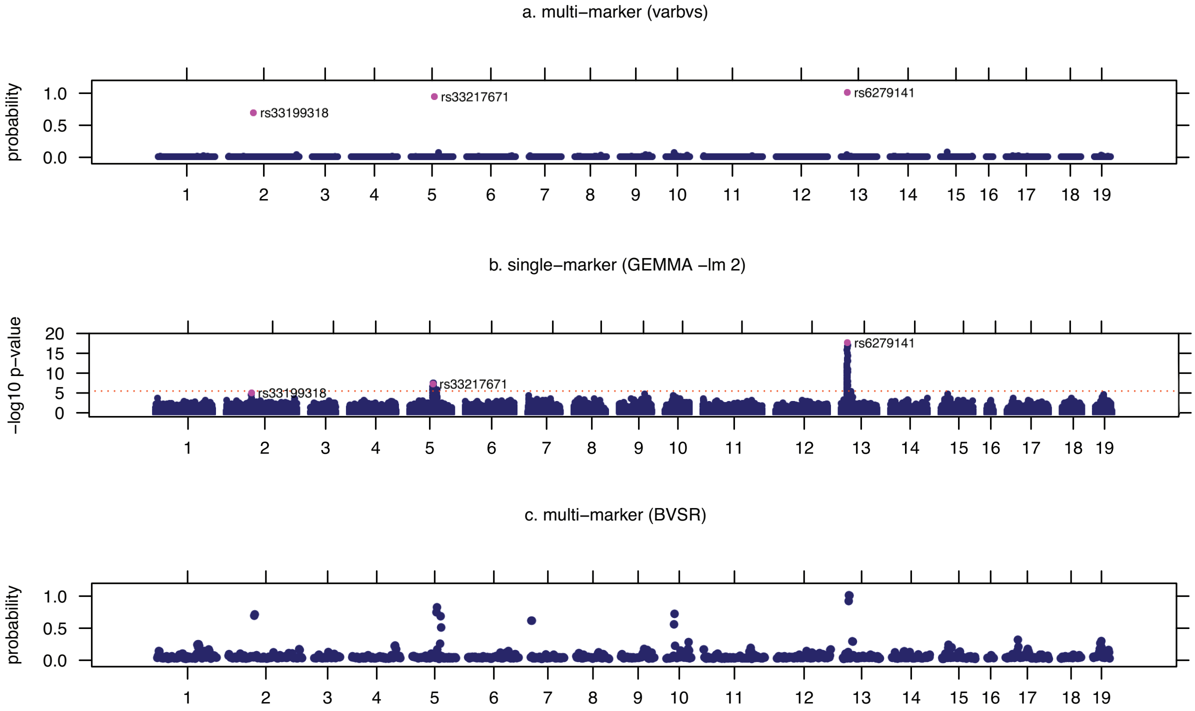

Figure 3: QTL mapping of testis weight in outbred mice. (a) Posterior inclusion probabilities for all 79,748 candidate SNPs on chromosomes 1–19 computed using `varbvs`. Posterior probabilities are obtained by averaging over `logodds` settings. SNPs with PIPs greater than 0.5 are highlighted. (b) $p$ values for the same candidate SNPs computed in GEMMA using the likelihood-ratio test, which ignores correlations between markers. The dotted red line gives the significance threshold determined via permutation analysis, at $p$ value $= 2 \times 10^{-6}$ (Parker *et al.* 2016). (c) Posterior probabilities computed using the BVSR method (Zhou *et al.* 2013). In BVSR, since multiple correlated SNPs at a single QTL are expected to be included in the model with lower probability, plotting individual PIPs does not highlight the testis weight QTLs. Therefore, we divide each chromosome into contiguous segments containing 100 SNPs, in which consecutive segments overlap by 50 SNPs, and show the posterior probability that at least 1 SNP is included within each of these segments. By definition, the segment-level probabilities are always larger than the inclusion probabilities for individual SNPs. All plots were generated using script `demo.cfw.R`.

of a GWAS. Reassuringly, the loci with the strongest support for association in the single-marker analysis (Fig. 3b) also exhibit the strongest support for association in the multi-marker analysis (Fig. 3a), and the SNPs included with the highest posterior probabilities are among the SNPs with the smallest $p$ values. Additionally, the testis weight QTL on chromosome 2 is not significant in the single-marker analysis ($p$ value $= 5.2 \times 10^{-6}$), yet shows moderate probability of association in the multi-marker analysis. The multi-marker association signal at this locus is concentrated in a small region that only contains one gene, *Myo3b*, providing an additional promising gene candidate for further investigation.

We also observe (Fig. 3b) that many SNPs have low $p$ values at each of the identified testis weight loci. This illustrates the common situation in GWAS in which many SNPs at a single locus are associated with the trait. In general, when multiple variables are correlated with each other, the fully-factorized variational approximation in **varbvs** tends to concentrate the posterior mass on a single variable. Often, this selected variable is the "best" possible choice—that is, the variable that explains the most variation in the outcome. Therefore, in GWAS, a SNP with a high PIP indicates (1) this SNP is probably associated with the trait, and (2) one or more nearby SNPs may be associated with the trait. In general, a high PIP for variable $X_i$ indicates one or more associations among all variables correlated with $X_i$.

We also assessed the impact of the fully-factorized variational approximation on the computation of posterior probabilities by comparing the **varbvs** results (Fig. 3a) against another BVS method, BVSR (Zhou *et al.* 2013), which is also implemented in the GEMMA software. BVSR uses more computationally intensive MCMC techniques to estimate posterior probabilities (Fig. 3c). Although these two methods aren't perfectly comparable due to differences in prior specification, posterior computation procedures and statistics used to summarize the association results, we observe that the testis weight loci with the strongest support in **varbvs** show a similar level of support in the BVSR analysis.

## 5. Example: mapping Crohn's disease risk loci

Our third example illustrates how **varbvs** can be applied to a very large genetic data set to map genetic loci contributing to complex disease risk. This is a much larger data set than the previous example, with 4,686 samples—1,748 Crohn's disease cases and 2,938 controls—and 442,001 SNPs (Wellcome Trust Case Control Consortium 2007). The genotypes of the cases and controls are stored in a $4,686 \times 442,001$ matrix `X`, and the binary outcome is case-control status:

```
> print(summary(factor(y)))

   0    1
2938 1748
```

We model case-control status using the logistic regression model (Equation 4), with the 442,001 SNPs as candidate variables, and no additional covariates. On a machine with a 2.5 GHz Intel Xeon CPU, fitting the BVS model to the data took 39 hours to complete with the following call:

```
R> fit <- varbvs(X, NULL, y, family = "binomial", logodds = seq(-6,-3,0.25))
```

In the R package, we have included a vignette, `demo.cd.R`, that reproduces all the results and plots. Unfortunately, the data set required to run the script, `cd.RData`, cannot be made publicly available due to data sharing restrictions, so readers wishing to reproduce this analysis must first apply for data access by contacting the Wellcome Trust Case Control Consortium. We present this example mainly to demonstrate the ability of **varbvs** to efficiently tackle very large data sets.

Similar to the previous examples, the fitted regression model is very sparse; only 8 out of the 442,001 candidate variables are included in the model with probability 0.5 or greater:

```
R> print(summary(fit,nv = 9))
```

```
Summary of fitted Bayesian variable selection model:
family:     binomial   num. hyperparameter settings: 13
samples:    4686       iid variable selection prior: yes
variables:  442001     fit prior var. of coefs (sa): yes
fit approx. factors (eta):    yes
maximum log-likelihood lower bound: -3043.2388
Hyperparameters:
        estimate Pr>0.95             candidate values
sa         0.032 [0.0201,0.04]       NA--NA
logodds    -4.06 [-4.25,-3.75]       (-6.00)--(-3.00)
Selected variables by probability cutoff:
>0.10 >0.25 >0.50 >0.75 >0.90 >0.95
   13    10     8     7     7     7
Top 9 variables by inclusion probability:
     index   variable  prob PVE  coef*  Pr(coef.>0.95)
  1  71850 rs10210302 1.000  NA -0.313 [-0.397,-0.236]
  2  10067 rs11805303 1.000  NA  0.291 [+0.207,+0.377]
  3 140044 rs17234657 1.000  NA  0.370 [+0.255,+0.484]
  4 381590 rs17221417 1.000  NA  0.279 [+0.192,+0.371]
  5 402183  rs2542151 0.992  NA  0.290 [+0.186,+0.392]
  6 271787 rs10995271 0.987  NA  0.236 [+0.151,+0.323]
  7 278438  rs7095491 0.969  NA  0.222 [+0.141,+0.303]
  8 168677  rs9469220 0.586  NA -0.194 [-0.269,-0.118]
  9  22989 rs12035082 0.485  NA  0.195 [+0.111,+0.277]
*See help(varbvs) about interpreting coefficients in logistic regression.
```

These results are also summarized in Fig. 4a. Comparing these results against the originally published association results (Wellcome Trust Case Control Consortium 2007), the 7 SNPs included in the regression model with probability greater than 0.9 correspond to loci identified with the smallest trend $p$ values, between $7.1 \times 10^{-14}$ and $2.68 \times 10^{-7}$. Additionally, in most cases the SNP the highest posterior probability is the exact same SNP with the smallest trend $p$ value. (See Carbonetto and Stephens 2013 for an extended comparison of the $p$ values and PIPs.) Only one disease locus, near gene *IRGM* on chromosome 5, has substantially stronger support in the single-marker analysis; the originally reported $p$ value is $5.1 \times 10^{-8}$, whereas the BVS analysis yields a largest posterior probability of 0.05 at this locus. This association has been reproduced in subsequent studies Barrett *et al.* (2008); Franke *et al.* (2010). We conclude that BVS yields strong support for nearly the same reported $p$ values at the established "whole-genome" significance threshold, $5 \times 10^{-7}$, and does so without having to settle on appropriate criterion for determining significance.

To further validate the **varbvs** analysis of the Crohn's disease data, we computed posterior probabilities using the BVSR method. Again, we obtain broadly similar variable selection results; the loci with the strongest support in the **varbvs** analysis (Fig. 4a) are the same loci identified by the BVSR method (Fig. 4b) aside from a few loci with moderate support in the BVSR analysis near genes *TNFSF18*, *MST1* and *IRGM*.

Revisiting the same Crohn's disease data set, we demonstrate the use of **varbvs** for model comparison to assess support for a biological hypothesis about disease risk. This model

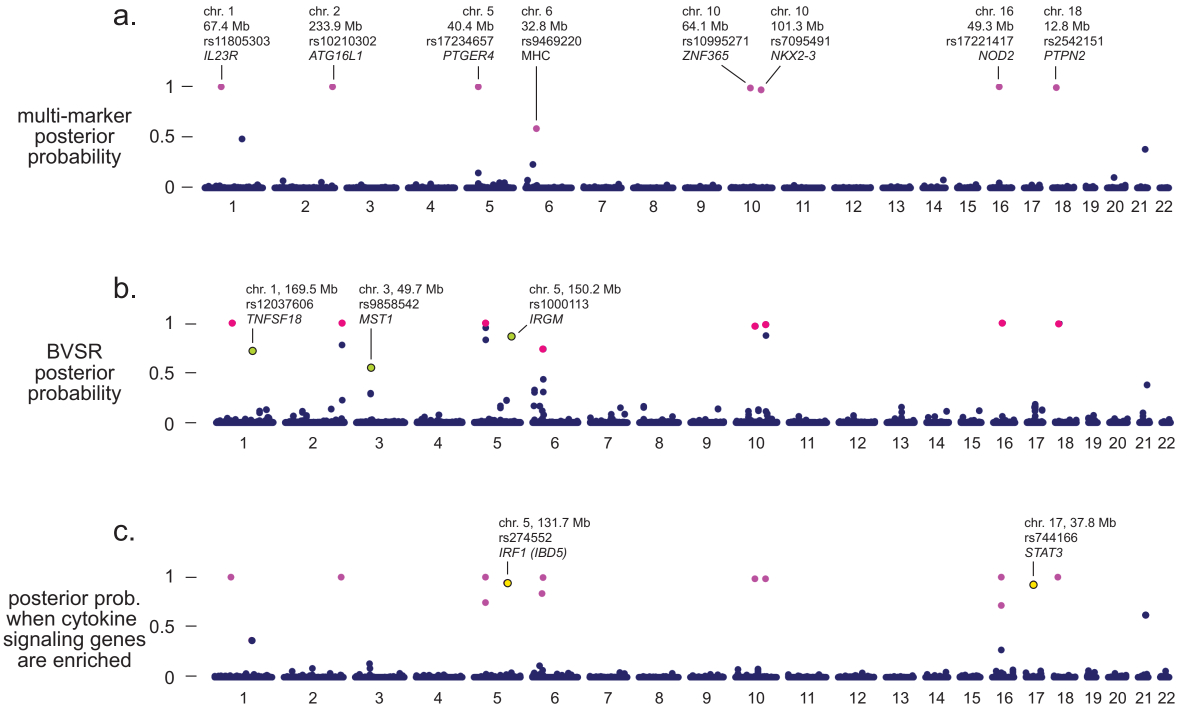

Figure 4: **varbvs** and BVSR analysis of Crohn's disease data. (a) Posterior inclusion probabilities for all 442,001 candidate SNPs on chromosomes 1–22. SNPs with PIP greater than 0.5 are highlighted and labeled. Candidate gene(s) near the included SNP are included with each label for illustration. All SNP base-pair positions are based on Human Genome Assembly hg17 (NCBI release 35). (b) Posterior probabilities estimated in BVSR (Zhou *et al.* 2013). Similar to the testis weight example (Sec. 4), each chromosome is divided into overlapping 50-SNP segments, and the plot shows the posterior probability that at least 1 SNP is included within each of the segments. The 3 segments with posterior probability greater than 0.5 in the BVSR analysis that do not contain a SNP with PIP greater than 0.5 in the **varbvs** analysis are highlighted in light green. (c) PIPs for all SNPs conditioned on enrichment of cytokine signaling genes. The two SNPs with PIP greater than 0.5 only after prioritizing SNPs near cytokine signaling genes are highlighted in yellow.

comparison analysis is implemented in script `demo.cytokine.R`. As before, the required data set, `cd.RData`, cannot be shared without prior approval. Here, we incorporate additional information about the 442,001 candidate variables, stored in a vector, `cytokine`:

```
R> print(summary(factor(cytokine)))

      0       1 
435290    6711 
```

We assigned `cytokine[i] = 1` if SNP `i` is one of the 6,711 SNPs located within 100 kb of a gene in the "Cytokine signaling in immune system" gene set; otherwise, we set `cytokine[i] = 0`. This gene set was identified in an interrogation of 3,158 pathways from 8 publicly available pathway databases (Carbonetto and Stephens 2013). [This is pathway id 75790 in the Reactome database (Croft *et al.* 2011), and pathway id 366171 in the BioSystems database (Geer *et al.* 2010).]

To assess the relevance of cytokine signaling genes to Crohn's disease susceptibility, we introduce a modified prior that allows for the 6,711 SNPs near cytokine signaling genes to be included in the model with higher probability, *a priori*:

```
R> logodds <- matrix(-4,442001,13)
R> logodds[cytokine == 1,] <- matrix(seq(0,3,0.25) - 4,6711,13,byrow = TRUE)
```

To simplify this example, we have fixed the "default" prior log-odds to `-4`, which is the maximum-likelihood setting under the exchangeable prior estimated in the analysis above. We consider 13 candidate settings of the prior for SNPs near cytokine signaling genes, ranging from `-4` (1 out of 10,000 SNPs is included, identical to SNPs not assigned to this gene set) to `-1` (1 out of 10 SNPs is included). We then fit the BVS model to the data using this modified prior:

```
R> fit.cytokine <- varbvs(X, NULL, y, family = "binomial", logodds = logodds)
```

The new variable selection results, averaging over all candidate settings of the prior log-odds, are summarized in Fig. 4c. The SNPs identified in the previous analysis are preserved under this new prior, and two new SNPs, near genes *IRF1* and *STAT3*, show strong support for association only after conditioning on enrichment of SNPs near cytokine signaling genes. Both of these genetic associations are corroborated by subsequent Crohn's disease GWAS with larger numbers of samples (Barrett *et al.* 2008; Franke *et al.* 2010; Jostins *et al.* 2012).

To assess support for this model, we compare against the "null" model in which all SNPs are equally likely to be included *a priori*:

```
R> fit.null <- varbvs(X, NULL, y, "binomial", logodds = -4)
R> BF <- bayesfactor(fit.null$logw, fit.cytokine$logw)
R> print(format(BF, scientific = TRUE))

[1] "9.355e+05"
```

This Bayes factor provides us with strong evidence for the contribution of cytokine signaling genes to Crohn's disease risk.

## 6. Summary and discussion

In this paper, we demonstrated the benefits of Bayesian variable selection techniques for regression analysis, and we showed how **varbvs** offers an easy-to-use interface for applying BVS to large data sets. In our examples, we explicitly hid the more technical details of BVS. For interested readers, additional mathematical details and derivations are found in the Appendix, and in Carbonetto and Stephens (2012). In the remainder of this paper, we provide some additional background on our methods and some general usage guidelines.

As we demonstrated in several of the examples, one benefit of BVS is that it provides a measure of uncertainty in the parameter estimates. It is also preferrable in most settings to account for uncertainty in the hyperparameters—that is, the parameters specifying the priors. However, in practice this is often not done because it requires careful selection of an additional set of priors for the hyperparameters. Therefore, we have provided default priors that are suitable in many settings, which allows the practitioner to expedite the analysis, and perhaps revisit the prior choices at a later date. The default priors are based on more detailed discussions from our earlier papers on this topic (Guan and Stephens 2011; Servin and Stephens 2007; Zhou *et al.* 2013). Our framework does not offer complete flexibility—we do not implement the $g$-prior, for example—so we aim to allow for a broader range of priors in future versions of **varbvs**. As an alternative, we allow for the option of computing a single point estimate of the hyperparameters, which has the benefit of much faster computation.

Fast computation of posterior probabilities is made possible in **varbvs** by the formulation of a variational approximation with the assumption that all coefficients are conditionally independent. Although this approximation will be poor when variables are correlated, *a key idea in this work is that this is typically not an issue so long as the variable selection results are used appropriately.* (Note that this issue is not unique to **varbvs**—many other variable selection methods should be used with similar prudence.) For example, consider the simple scenario in which $p = 2$ candidate variables are closely correlated, and one of them explains the outcome with probability close to 1. Under the correct posterior distribution, we would expect that each variable is included with probability about 0.5. The variational approximation, due to the conditional independence assumption, will typically get this wrong, and concentrate most of the posterior weight on one variable (the actual variable that is chosen will depend on the starting conditions of the optimization). While the individual PIPs are incorrect, a statistic summarizing the variable selection for both correlated variables (e.g., the total number of variables included in the model) is expected to be accurate. In general, if variables can be reasonably grouped together based on their correlations, we recommend interpreting the variable selection results at a group level. We illustrated this approach in the two GWAS examples: a SNP with a high PIP indicates that this SNP is probably associated with the trait, and one or more nearby SNPs within a chromosomal region, or "locus," may be associated as well. Therefore, we interpreted the GWAS variable selection results at the level of loci, rather than at the level of individual SNPs. Also, in general it is safer to assess support for an association by computing the posterior probability that at least 1 variable within a given group is included in the model, which is easily computed from the `varbvs` output (see Sec. 3.5). This is more reliable as the posterior mass may be spread out among multiple correlated variables, in which case a single PIP would not immediately stand out (e.g., using the `plot` function).

In practice, the final estimates can be sensitive to initialization of the variational parameters. We have reduced this sensitivity by including an additional optimization step that first identifies a good initialization of the variational parameters. However, it is good practice to verify that different random initializations of these parameters do not yield substantially different conclusions.

The computational complexity of the co-ordinate ascent algorithm for fitting the variational approximation is linear in the number of samples and in the number of variables so long as the correlations between variables are mostly small. This makes the algorithm paticularly suitable for GWAS since correlations are limited by recombination. However, for data sets with widespread correlations between variables, convergence of the algorithm can be unreasonably slow. We are currently investigating faster alternatives using quasi-Newton methods and

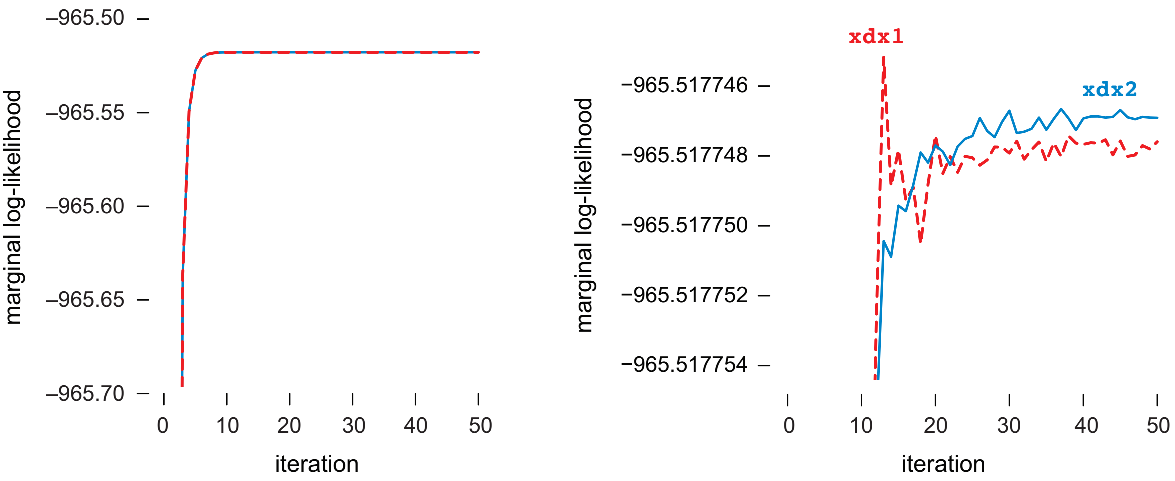

Figure 5: Convergence of **varbvs** model fitting algorithm with less numerically stable (`xdx1`) and more numerically stable (`xdx2`) updates. The vertical axes show the variational lower bound to the marginal log-likelihood, which also serves as the objective to be maximized; the co-ordinate ascent updates terminate when they no longer increase the lower bound. The right-hand plot is a magnified version of the left-hand plot.

acceleration schemes such as SQUAREM (Varadhan and Roland 2008; Varadhan 2016).

Since BVS methods require that the entire data set be available in memory, memory is currently the most important limiting factor for applying **varbvs** to large data sets. While this limitation is not easily circumvented, distributed computing frameworks such as Spark Zaharia *et al.* (2012) could potentially be used to overcome this barrier.

Finally, we would like to remark on an often overlooked aspect of statistical analyses that can be especially critical for large-scale data sets—numerical stability. For the logistic regression model, part of the variational optimization algorithm involves computing the diagonal entries of of the matrix product $\mathbf{X}^T\hat{D}\mathbf{X}$, in which $\hat{D}$ is an $n \times n$ diagonal matrix with diagonal entries defined by the variational approximation (see Appendix B). In the MATLAB implementation (the R code is similar but a bit harder to follow), the following two lines of code are mathematically equivalent,

```
xdx1 = diag(X'*D*X) - (X'*d).^2/sum(d)
xdx2 = diag(X'*D*X) - (X'*(d/sqrt(sum(d)))).^2
```

where `d = diag(D)`, yet in floating-point arithmetic, the order of operations affects the numerical precision of the final result, which can in turn affect the stability of the co-ordinate ascent updates. To illustrate this, we applied **varbvs**, using the two different updates (`xdx1` and `xdx2`), to a data set with simulated variables and a binary outcome. In Fig. 5, we see that the second update (`xdx2`), corresponding to the solid blue line in the plots, yields iterates that progress more smoothly to a stationary point of the objective function, whereas the first update (`xdx1`) terminated prematurely in this example because it produced a large decrease in the objective. This small example illustrates the more general point that the numerical stability of operations can have a surprisingly large effect the quality of the final solution, particularly in large data sets.

## Acknowledgments

Thanks to John Zekos for computing support. Thanks to Abraham Palmer, Clarissa Parker, Shyam Gopalakrishnan, Arimantas Lionikas, and other present and former members of the Palmer Lab for their contributions to the mouse data set. Thanks to Xiang Zhu for his feedback on the code. Thanks to Cisca Wijmenga and Gosia Trynka (and others?) for their assistance with the celiac study data (only if I include the celiac disease example). Thank you to Karl Broman, Hadley Wickham and the authors of **glmnet** (Jerome Friedman, Trevor Hastie, Noah Simon, Rob Tibshirani) for providing examples of excellent R packages and code that have influenced the design of the varbvs interface. Thanks to Ravi Varadhan and Yu Du for their feedback on the formatting and R examples in the paper. Thanks to Ann Carbonetto for her support and encouragement.

## References

Attias H (1999). "Independent Factor Analysis." *Neural Computation*, **11**(4), 803–851.

Barrett JC, Hansoul S, Nicolae DL, Cho JH, Duerr RH, Rioux JD, Brant SR, Silverberg MS, Taylor KD, Barmada MM, Bitton A, Dassopoulos T, Datta LW, Green T, Griffiths AM, Kistner EO, Murtha MT, Regueiro MD, Rotter JI, Schumm LP, Steinhart AH, Targan SR, Xavier RJ, Libioulle C, Sandor C, Lathrop M, Belaiche J, Dewit O, Gut I, Heath S, Laukens D, Mni M, Rutgeerts P, Van Gossum A, Zelenika D, Franchimont D, Hugot JP, de Vos M, Vermeire S, Louis E, Cardon LR, Anderson CA, Drummond H, Nimmo E, Ahmad T, Prescott NJ, Onnie CM, Fisher SA, Marchini J, Ghori J, Bumpstead S, Gwilliam R, Tremelling M, Deloukas P, Mansfield J, Jewell D, Satsangi J, Mathew CG, Parkes M, Georges M, Daly MJ (2008). "Genome-wide Association Defines More Than 30 Distinct Susceptibility Loci for Crohn's Disease." *Nature Genetics*, **40**(8), 955–962.

Blei DM, Kucukelbir A, McAuliffe JD (2016). "Variational Inference: A Review for Statisticians." arXiv:1601.00670v3.

Bottolo L, Richardson S (2010). "Evolutionary Stochastic Search for Bayesian Model Exploration." *Bayesian Analysis*, **5**(3), 583–618.

Breheny P, Huang J (2011). "Coordinate Descent Algorithms for Nonconvex Penalized Regression, with Applications to Biological Feature Selection." *Annals of Applied Statistics*, **5**(1), 232–253.

Burden R, Faires JD (2005). *Numerical Analysis*. Thomson Brooks/Cole.

Carbonetto P, Stephens M (2012). "Scalable Variational Inference for Bayesian Variable Selection in Regression, and Its Accuracy in Genetic Association Studies." *Bayesian Analysis*, **7**(1), 73–108.

Carbonetto P, Stephens M (2013). "Integrated Enrichment Analysis of Variants and Pathways in Genome-wide Association Studies Indicates Central Role for IL-2 Signaling Genes in Type 1 Diabetes, and Cytokine Signaling Genes in Crohn's Disease." *PLoS Genetics*, **9**(10), e1003770.

Carbonetto P, Stephens M (2016). *varbvs: Large-Scale Bayesian Variable Selection Using Variational Methods*. URL http://CRAN.R-project.org/package=varbvs.

Chipman H, George EI, McCulloch RE (2001). "The Practical Implementation of Bayesian Model Selection." In *Model Selection*, volume 38 of *IMS Lecture Notes*, pp. 65–116.

Clyde MA, Ghosh J, Littman ML (2011). "Bayesian Adaptive Sampling for Variable Selection and Model Averaging." *Journal of Computational and Graphical Statistics*, **20**(1), 80–101.

Croft D, O'Kelly G, Wu G, Haw R, Gillespie M, Matthews L, Caudy M, Garapati P, Gopinath G, Jassal B, Jupe S, Kalatskaya I, Mahajan S, May B, Ndegwa N, Schmidt E, Shamovsky V, Yung C, Birney E, Hermjakob H, D'Eustachio P, Stein L (2011). "Reactome: A Database of Reactions, Pathways and Biological Processes." *Nucleic Acids Research*, **39**(S1), D691–D697.

Dellaportas P, Forster JJ, Ntzoufras I (2002). "On Bayesian Model and Variable Selection Using MCMC." *Statistics and Computing*, **12**(1), 27–36.

Dettling M (2004). "BagBoosting for Tumor Classification with Gene Expression Data." *Bioinformatics*, **20**(18), 3583–3593.

Erbe M, Hayes BJ, Matukumalli LK, Goswami S, Bowman PJ, Reich CM, Mason BA, Goddard ME (2012). "Improving Accuracy of Genomic Predictions Within and Between Dairy Cattle breeds with Imputed High-density Single Nucleotide Polymorphism Panels." *Journal of Dairy Science*, **95**(7), 4114–4129.

Franke A, McGovern DPB, Barrett JC, Wang K, Radford-Smith GL, Ahmad T, Lees CW, Balschun T, Lee J, Roberts R, Anderson CA, Bis JC, Bumpstead S, Ellinghaus D, Festen EM, Georges M, Green T, Haritunians T, Jostins L, Latiano A, Mathew CG, Montgomery GW, Prescott NJ, Raychaudhuri S, Rotter JI, Schumm P, Sharma Y, Simms LA, Taylor KD, Whiteman D, Wijmenga C, Baldassano RN, Barclay M, Bayless TM, Brand S, Buning C, Cohen A, Colombel JF, Cottone M, Stronati L, Denson T, De Vos M, D'Inca R, Dubinsky M, Edwards C, Florin T, Franchimont D, Gearry R, Glas J, Van Gossum A, Guthery SL, Halfvarson J, Verspaget HW, Hugot JP, Karban A, Laukens D, Lawrance I, Lemann M, Levine A, Libioulle C, Louis E, Mowat C, Newman W, Panes J, Phillips A, Proctor DD, Regueiro M, Russell R, Rutgeerts P, Sanderson J, Sans M, Seibold F, Steinhart AH, Stokkers PCF, Torkvist L, Kullak-Ublick G, Wilson D, Walters T, Targan SR, Brant SR, Rioux JD, D'Amato M, Weersma RK, Kugathasan S, Griffiths AM, Mansfield JC, Vermeire S, Duerr RH, Silverberg MS, Satsangi J, Schreiber S, Cho JH, Annese V, Hakonarson H, Daly MJ, Parkes M (2010). "Genome-wide Meta-analysis Increases to 71 the Number of Confirmed Crohn's Disease Susceptibility Loci." *Nature Genetics*, **42**(12), 1118–1125.

Friedman J, Hastie T, Höfling H, Tibshirani R (2007). "Pathwise Coordinate Optimization." *Annals of Applied Statistics*, **2**, 302–332.

Friedman J, Hastie T, Tibshirani R (2010). "Regularization Paths for Generalized Linear Models via Coordinate Descent." *Journal of Statistical Software*, **33**(1), 1–22. URL http://www.jstatsoft.org/v033/i01.

Geer LY, Marchler-Bauer A, Geer RC, Han L, He J, He S, Liu C, Shi W, Bryant SH (2010). "The NCBI BioSystems Database." *Nucleic Acids Research*, **38**(S1), D492–D496.

George EI (2000). “The Variable Selection Problem.” *Journal of the American Statistical Association*, **95**(452), 1304–1308.

George EI, McCulloch RE (1993). “Variable Selection via Gibbs Sampling.” *Journal of the American Statistical Association*, **88**(423), 881–889.

Golub TR, Slonim DK, Tamayo P, Huard C, Gaasenbeek M, Mesirov JP, Coller H, Loh ML, Downing JR, Caligiuri MA, Bloomfield CD, Lander ES (1999). “Molecular Classification of Cancer: Class Discovery and Class Prediction by Gene Expression Monitoring.” *Science*, **286**(5439), 531–537.

Guan Y, Stephens M (2011). “Bayesian Variable Selection Regression for Genome-wide Association Studies, and Other Large-scale Problems.” *Annals of Applied Statistics*, **5**(3), 1780–1815.

Heskes T, Zoeter O, Wiegerinck W (2004). “Approximate Expectation Maximization.” In S Thrun, LK Saul, B Schölkopf (eds.), *Advances in Neural Information Processing Systems 16*, pp. 353–360.

Hoeting JA, Madigan D, Raftery AE, Volinsky CT (1999). “Bayesian Model Averaging: A Tutorial.” *Statistical Science*, **14**(4), 382–401.

Hoggart CJ, Whittaker JC, De Iorio M, Balding DJ (2008). “Simultaneous Analysis of All SNPs in Genome-wide and Re-sequencing Association Studies.” *PLoS Genetics*, **7**(4), e1000130.

Jaakkola TS, Jordan MI (2000). “Bayesian Parameter Estimation via Variational Methods.” *Statistics and Computing*, **10**(1), 25–37.

Jefferys WH, Berger JO (1992). “Ockham’s Razor and Bayesian analysis.” *American Scientist*, **80**(1), 64–72.

Jordan MI, Ghahramani Z, Jaakkola TS, Saul LK (1999). “An Introduction to Variational Nethods for Graphical Models.” *Machine Learning*, **37**(2), 183–233.

Jostins L, Ripke S, Weersma RK, Duerr RH, McGovern DP, Hui KY, Lee JC, Philip Schumm L, Sharma Y, Anderson Ca, Essers J, Mitrovic M, Ning K, Cleynen I, Theatre E, Spain SL, Raychaudhuri S, Goyette P, Wei Z, Abraham C, Achkar JP, Ahmad T, Amininejad L, Ananthakrishnan AN, Andersen V, Andrews JM, Baidoo L, Balschun T, Bampton Pa, Bitton A, Boucher G, Brand S, Büning C, Cohain A, Cichon S, D’Amato M, De Jong D, Devaney KL, Dubinsky M, Edwards C, Ellinghaus D, Ferguson LR, Franchimont D, Fransen K, Gearry R, Georges M, Gieger C, Glas J, Haritunians T, Hart A, Hawkey C, Hedl M, Hu X, Karlsen TH, Kupcinskas L, Kugathasan S, Latiano A, Laukens D, Lawrance IC, Lees CW, Louis E, Mahy G, Mansfield J, Morgan AR, Mowat C, Newman W, Palmieri O, Ponsioen CY, Potocnik U, Prescott NJ, Regueiro M, Rotter JI, Russell RK, Sanderson JD, Sans M, Satsangi J, Schreiber S, Simms La, Sventoraityte J, Targan SR, Taylor KD, Tremelling M, Verspaget HW, De Vos M, Wijmenga C, Wilson DC, Winkelmann J, Xavier RJ, Zeissig S, Zhang B, Zhang CK, Zhao H, Silverberg MS, Annese V, Hakonarson H, Brant SR, Radford-Smith G, Mathew CG, Rioux JD, Schadt EE, Daly MJ, Franke A, Parkes M, Vermeire S, Barrett JC, Cho JH (2012). “Host-microbe Interactions Have Shaped the Genetic Architecture of Inflammatory Bowel Disease.” *Nature*, **491**(7422), 119–124.

Kang HM, Sul JH, Service SK, Zaitlen NA, Kong SY, Freimer NB, Sabatti C, Eskin E (2010). "Variance Component Model to Account for Sample Structure in Genome-wide Association Studies." *Nature Genetics*, **42**(4), 348–354.

Kass RE, Raftery AE (1995). "Bayes Factors." *Journal of the American Statistical Association*, **90**(430), 773–795.

Lee SH, van der Werf JHJ, Hayes BJ, Goddard ME, Visscher PM (2008). "Predicting Unobserved Phenotypes for Complex Traits from Whole-genome SNP Data." *PLoS Genetics*, **4**, e1000231.

Liang F, Paulo R, Molina G, Clyde MA, Berger JO (2008). "Mixtures of g Priors for Bayesian Variable Selection." *Journal of the American Statistical Association*, **103**(481), 410–423.

Listgarten J, Lippert C, Kadie CM, Davidson RI, Eskin E, Heckerman D (2012). "Improved Linear Mixed Models for Genome-wide Association Studies." *Nature Methods*, **9**(6), 525–526.

Logsdon BA, Hoffman GE, Mezey JG (2010). "A Variational Bayes Algorithm for Fast and Accurate Multiple Locus Genome-wide Association Analysis." *BMC Bioinformatics*, **11**, 58.

MacKay DJC (1992). "Bayesian Interpolation." *Neural Computation*, **4**(3), 415–447.

Mendis SHS, Meachem SJ, Sarraj MA, Loveland KL (2011). "Activin A Balances Sertoli and Germ Cell Proliferation in the Fetal Mouse Testis." *Biology of Reproduction*, **84**(2), 379–391.

Meuwissen THE, Hayes B, Goddard M (2001). "Prediction of Total Genetic Value Using Genome-wide Dense Marker Maps." *Genetics*, **157**(4), 1819–1829.

Mitchell TJ, Beauchamp JJ (1988). "Bayesian Variable Selection in Linear Regression." *Journal of the American Statistical Association*, **83**(404), 1023–1032.

Mithraprabhu S, Mendis S, Meachem SJ, Tubino L, Matzuk MM, Brown CW, Loveland KL (2010). "Activin Bioactivity Affects Germ Cell Differentiation in the Postnatal Mouse Testis In Vivo." *Biology of Reproduction*, **82**(5), 980–990.

Moser G, Lee SH, Hayes BJ, Goddard ME, Wray NR, Visscher PM (2015). "Simultaneous Discovery, Estimation and Prediction analysis of Complex Traits Using a Bayesian Mixture Model." *PLOS Genetics*, **11**(4), e1004969.

Neal R, , Hinton G (1998). "A View of the EM Algorithm that Justifies Incremental, Sparse, and Other Variants." In M Jordan (ed.), *Learning in Graphical Models*, pp. 355–368. Kluwer Academic Publishers, Dordrecht.

O'Brien SM, Dunson DB (2004). "Bayesian Multivariate Logistic Regression." *Biometrics*, **60**(3), 739–746.

O'Hara RB, Sillanpäa MJ (2009). "A Review of Bayesian Variable Selection Methods: What, How and Which." *Bayesian Analysis*, **4**(1), 85–117.

Ormerod JT, Wand MP (2010). "Explaining Variational Approximations." *The American Statistician*, **64**(2), 140–153.

Parker CC, Gopalakrishnan S, Carbonetto P, Gonzales NM, Leung E, Park YJ, Aryee E, Davis J, Blizard DA, Ackert-Bicknell CL, Lionikas A, Pritchard JK, Palmer AA (2016). "Genome-wide Association Study of Behavioral, Physiological and Gene Expression Traits in Outbred CFW Mice." *Nature Genetics*, **48**(8), 919–926.

Perez P, de los Campos G (2014). "Genome-Wide Regression and Prediction with the BGLR Statistical Package." *Genetics*, **198**(2), 483–495.

Purcell S, Neale B, Todd-Brown K, Thomas L, Ferreira MAR, Bender D, Maller J, Sklar P, de Bakker PI, Daly MJ, Sham PC (2007). "PLINK: A Tool Set for Whole-genome Association and Population-based Linkage Analyses." *American Journal of Human Genetics*, **81**(3), 559–575.

R Core Team (2016). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. URL https://www.R-project.org.

Servin B, Stephens M (2007). "Imputation-based Analysis of Association Studies: Candidate Regions and Quantitative Traits." *PLoS Genetics*, **3**(7), 1296–1308.

The MathWorks, Inc (2016). *MATLAB: The Language of Technical Computing, Version R2016a*. The MathWorks, Inc., Natick, Massachusetts. URL http://www.mathworks.com/products/matlab.

Tibshirani R (1994). "Regression Selection and Shrinkage via the Lasso." *Journal of the Royal Statistical Society Series B*, **58**(1), 267–288.

Tibshirani R, Saunders M, Rosset S, Zhu J, Knight K (2005). "Sparsity and Smoothness via the Fused Lasso." *Journal of the Royal Statistical Society Series B*, **67**(1), 91–108.

Tomaszewski J, Joseph A, Archambeault D, Yao HCH (2007). "Essential Roles of Inhibin Beta A in Mouse Epididymal Coiling." *Proceedings of the National Academy of Sciences*, **104**(27), 11322–11327.

Varadhan R (2016). *SQUAREM: Squared Extrapolation Methods for Accelerating EM-Like Monotone Algorithms*. URL http://CRAN.R-project.org/package=SQUAREM.

Varadhan R, Roland C (2008). "Simple and Globally Convergent Methods for Accelerating the Convergence of any EM Algorithm." *Scandinavian Journal of Statistics*, **35**(2), 335–353.

Wainwright MJ, Jordan MI (2008). "Graphical Models, Exponential Families, and Variational Inference." *Foundations and Trends in Machine Learning*, **1**(1–2), 1–305.

Wallace C, Cutler AJ, Pontikos N, Pekalski ML, Burren OS, Cooper JD, García AR, Ferreira RC, Guo H, Walker NM, Smyth DJ, Rich SS, Onengut-Gumuscu S, Sawcer SJ, Ban M, Richardson S, Todd JA, Wicker LS (2015). "Dissection of a Complex Disease Susceptibility Region Using a Bayesian Stochastic Search Approach to Fine Mapping." *PLOS Genetics*, **11**(6), e1005272.

Wellcome Trust Case Control Consortium (2007). "Genome-wide Association Study of 14,000 Cases of Seven Common Diseases and 3,000 Shared Controls." *Nature*, **447**(7145), 661–678.

Yang J, Lee SH, Goddard ME, Visscher PM (2011). "GCTA: A Tool for Genome-wide Complex Trait Analysis." *American Journal of Human Genetics*, **88**(1), 76–82.

Zaharia M, Chowdhury M, Das T, Dave A, Ma J, McCauley M, Franklin MJ, Shenker S, Stoica I (2012). "Resilient Distributed Datasets: A Fault-tolerant Abstraction for In-memory Cluster Computing." In *Proceedings of the 9th USENIX Conference on Networked Systems Design and Implementation*.

Zellner A (1986). "On Assessing Prior Distributions and Bayesian Regression Analysis with g-prior Distributions." In PK Goal, A Zellner (eds.), *Bayesian Inference and Decision Techniques: Essays in Honor of Bruno de Finetti*, pp. 233–243. Edward Elgar Pub. Ltd.

Zhou X, Carbonetto P, Stephens M (2013). "Polygenic Modeling with Bayesian Sparse Linear Mixed Models." *PLoS Genetics*, **9**(2), e1003264.

Zhou X, Stephens M (2012). "Genome-wide Efficient Mixed-model Analysis for Association Studies." *Nature Genetics*, **44**(7), 821–824.

Zou H, Hastie T (2005). "Regularization and Variable Selection via the Elastic Net." *Journal of the Royal Statistical Society Series B*, **67**(2), 301–320.

# A. Additional derivations for linear regression model

Most of the derivations for the linear regression model are given in Carbonetto and Stephens (2012). Here, we extend the variational approximation to allow for additional variables $(Z_1, \ldots, Z_m)^T$ that are included in the model with probability 1, and a non-exchangeable prior on the regression coefficients $\beta_i$. Additionally, we derive an approximate EM algorithm for the residual variance $\sigma^2$ and prior variance $\sigma_a^2$.

First, we analytically integrate out the regression coefficients $u = (u_1, \ldots, u_m)^T$ by making use of the following result:

$$|\Sigma_0|^{1/2} \, \Pr(y \,|\, \mathbf{X}, \mathbf{Z}, \beta, \sigma^2) = |\mathbf{Z}^T\mathbf{Z}|^{-1/2} \, \Pr(\hat{y} \,|\, \hat{\mathbf{X}}, \beta, \sigma^2), \tag{8}$$

in which $\Pr(y \,|\, \mathbf{X}, \mathbf{Z}, \beta, \sigma^2)$ is the multivariate normal likelihood defined by the linear regression model (Equation 3), $\Pr(\hat{y} \,|\, \hat{\mathbf{X}}, \beta, \sigma^2)$ is the likelihood given by linear regression $\hat{y} = \hat{\mathbf{X}}\beta + \sigma^2$, $u$ is assigned a multivariate normal prior with zero mean and covariance $\Sigma_0$ such that $|\Sigma_0^{-1}|$ is close to zero (yielding a "flat" prior density on $u$), and we define $\hat{\mathbf{X}} = \mathbf{X} - \mathbf{Z}(\mathbf{Z}^T\mathbf{Z})^{-1}\mathbf{Z}^T\mathbf{X}$ and $\hat{y} = y - \mathbf{Z}(\mathbf{Z}^T\mathbf{Z})^{-1}\mathbf{Z}^T y$. Therefore, we can easily account for the linear effects of covariates $Z$ by replacing all instances of $\mathbf{X}$ with $\hat{\mathbf{X}}$ and all instances of $y$ with $\hat{y}$, and by multipling the likelihood by $|\mathbf{Z}^T\mathbf{Z}|^{-1/2}$. Therefore, in the derivations below we assume the simpler linear regression $y = \mathbf{X}\beta + \sigma^2$, replace $X$ with $\hat{X}$ and $y$ with $\hat{y}$, and multiply by $|\mathbf{Z}^T\mathbf{Z}|^{-1/2}$ to obtain the final solution.

The basic idea behind the variational approximation is to formulate a lower bound to the marginal likelihood, $\Pr(y \,|\, \mathbf{X}, \theta) \geq e^{f(\mathbf{X}, y, \theta, \phi)}$, then to adjust the free parameters, which we denote here by $\phi \equiv \{\alpha, \mu, s\}$, so that this bound is as tight as possible. This lower bound is formulated by introducing a probability distribution $q(\beta;\phi)$ that approximates the posterior of $\beta$ given $\theta$. Maximizing the lower bound corresponds to finding the approximating distribution that best matches the posterior; more precisely, it amounts to searching for the free parameters $\phi$ that minimize the Kullback-Leibler divergence between $q(\beta;\phi)$ and the posterior of $\beta$ given $\theta$ (Jordan *et al.* 1999).

The "fully-factorized" class of approximating distributions yields the following analytical expression for the variational lower bound:

$$\begin{aligned} f(\mathbf{X}, y, \theta, \phi) = {} & -\frac{n}{2}\log(2\pi\sigma^2) - \frac{\|y - \mathbf{X}r\|_2^2}{2\sigma^2} - \frac{1}{2\sigma^2}\sum_{i=1}^p (\mathbf{X}^T\mathbf{X})_{ii}\mathrm{Var}[\beta_i] \\ & - \sum_{i=1}^p \alpha_i \log\left(\frac{\alpha_i}{\pi_i}\right) - \sum_{i=1}^p (1-\alpha_i)\log\left(\frac{1-\alpha_i}{1-\pi_i}\right) \\ & + \sum_{i=1}^p \frac{\alpha_i}{2}\left[1 + \log\left(\frac{s_i^2}{\sigma_a^2\sigma^2}\right) - \frac{s_i^2+\mu_i^2}{\sigma_a^2\sigma^2}\right], \end{aligned} \tag{9}$$

where $\|\cdot\|_2$ is the Euclidean norm, $r$ is a column vector with entries $r_i = \alpha_i\mu_i$, and $\mathrm{Var}[\beta_i] = \alpha_i(s_i^2 + \mu_i^2) - (\alpha_i\mu_i)^2$ is the variance of $i$th coefficient under the approximating distribution. As in Carbonetto and Stephens (2012), the co-ordinate updates for the free parameters conditioned on a hyperparameter setting $\theta$ are obtained by taking partial derivatives of the lower bound (Equation 9), setting these partial derivatives to zero, and solving for the free parameters. This yields the following expressions:

$$\mu_i = \tfrac{s_i^2}{\sigma^2}\big((\mathbf{X}^T y)_i - \textstyle\sum_{j\neq i}(\mathbf{X}^T\mathbf{X})_{ij}\alpha_j\mu_j\big) \tag{10}$$

$$s_i^2 = \sigma^2/\big((\mathbf{X}^T\mathbf{X})_{ii} + 1/\sigma_a^2\big) \tag{11}$$

$$\frac{\alpha_i}{1-\alpha_i} = \frac{\pi_i}{1-\pi_i} \times \frac{s_i}{\sigma\sigma_a} \times e^{\mu_i^2/(2s_i^2)}. \tag{12}$$

The E and M steps in the EM algorithm can be viewed as both minimizing the Kullback-Leibler divergence (Neal *et al.* 1998) or, equivalently in this case, maximizing the lower bound (Equation 9). Therefore, we obtain an "approximate" EM algorithm (e.g., Heskes *et al.* 2004) by computing posterior expectations in the E-step under the assumption that the true posterior is "fully-factorized." We derive the M-step updates for $\sigma^2$ and $\sigma_a^2$ in the standard way by solving for roots $\sigma^2$ and $\sigma_a^2$ of the gradient, yielding

$$\sigma^2 = \frac{\|y - \mathbf{X}r\|_2^2 + \sum_{i=1}^p(\mathbf{X}^T\mathbf{X})_{ii}\mathrm{Var}[\beta_i] + \sum_{i=1}^p \alpha_i(s_i^2+\mu_i^2)/\sigma_a^2)}{n + \sum_{i=1}^p \alpha_i} \tag{13}$$

$$\sigma_a^2 = \frac{\sum_{i=1}^p \alpha_i(s_i^2+\mu_i^2)}{\sigma^2\sum_{i=1}^p \alpha_i}. \tag{14}$$

# B. Additional derivations for logistic regression model

In the Appendix of Carbonetto and Stephens (2012), we described an extension to the fully-factorized variational approximation for Bayesian variable selection with a logistic regression model and an intercept. Here, we extend these derivations to allow for for additional variables $Z = (Z_1, \ldots, Z_m)^T$ that are not subject to the spike-and-slab priors.

We split the derivation into four parts: in the first part, we derive a linear approximation to the non-linear likelihood; in the second part, we analytically integrate out the coefficients $u$ from the linearized likelihood; in the third part, we introduce the fully-factorized variational approximation, and derive the co-ordinate ascent updates for maximizing the variational lower bound; finally, in the fourth part, we derive "M-step" updates for the additional free parameters $\eta_i$ that were introduced to approximate the logistic regression likelihood.

**Taking care of the nonlinear factors in the likelihood.** For the moment, we assume the simpler logistic regression with no additional variables $Z$; it is easy to introduce these variables into the expressions later on by substituting $\beta$ with $\binom{u}{\beta}$ and $\mathbf{X}$ with $(\mathbf{Z}\,\mathbf{X})$. The expression for the log-likelihood given the simpler logistic regression can be written as

$$\log \Pr(y \,|\, \mathbf{X}, \beta) = (y-1)^T \mathbf{X}\beta + \sum_{i=1}^n \log p_i, \tag{15}$$

in which we define $p_i \equiv \Pr(y_i = 1 \,|\, x_{i1}, \ldots, x_{ip}, \beta) = \sigma(\sum_{j=1}^p x_{ij}\beta_j)$, and $\sigma(x) = 1/(1+e^{-x})$ is the sigmoid function (or inverse of logit function). Written in this way, the linear components are contained exclusively in the first term of Equation 15.

The basic idea behind the variational approximation is to formulate a lower bound to the logarithm of the sigmoid function. Skipping the technical details (Jaakkola and Jordan 2000), we obtain the following lower bound:

$$\log \sigma(x) \geq \log \sigma(\eta) + \tfrac{1}{2}(x-\eta) - \tfrac{d}{2}(x^2 - \eta^2), \tag{16}$$

in which we define $d = \frac{1}{\eta}(\sigma(\eta) - \frac{1}{2})$. Notice that this expression introduces an additional parameter, $\eta$. This identity holds for any choice of $\eta$, and this is the free parameter that we will adjust to tighten the fit of the lower bound as best as possible. We will have one free parameter $\eta_i$ for every factor in the likelihood. Also notice that all terms involving $x$—later replaced by linear combinations of $\beta$—are linear or quadratic in $x$.

Inserting this lower bound into the expression for the log-likelihood, we obtain a lower bound to the log-likelihood, denoted by $g(\beta;\eta)$:

$$g(\beta;\eta) = \sum_{i=1}^n \log \sigma(\eta_i) + \tfrac{\eta_i}{2}(d_i\eta_i - 1) + (y - \tfrac{1}{2})^T \mathbf{X}\beta - \tfrac{1}{2}\beta^T \mathbf{X}^T D \mathbf{X}\beta, \tag{17}$$

where $D$ is the $n \times n$ matrix with diagonal entries $d_i$. By extension, we have a lower bound on the marginal likelihood:

$$\begin{aligned} \Pr(y \,|\, \mathbf{X}) &= \textstyle\int \Pr(y \,|\, \mathbf{X}, \beta)\, \Pr(\beta)\, d\beta \\ &\geq \textstyle\int e^{g(\beta;\eta)}\, \Pr(\beta)\, d\beta. \end{aligned} \tag{18}$$

**Integrating out the coefficients $u$.** Since we have assigned a normal prior to $u$ (with large variance), we can analytically integrate out $u$ from the lower bound (Equation 17), in which we substitute $\beta$ with $\binom{u}{\beta}$, and we substitute $\mathbf{X}$ with $(\mathbf{Z}\,\mathbf{X})$. This yields the following expression for the lower bound:

$$|\Sigma_0|^{1/2} \Pr(y \,|\, \mathbf{X}, \mathbf{Z}) \geq |\hat{\Sigma}|^{1/2} \textstyle\int e^{g^*(\beta;\eta)}\, \Pr(\beta)\, d\beta,$$

in which we define

$$g^*(\beta;\eta) = \sum_{i=1}^n \log\sigma(\eta_i) + \tfrac{\eta_i}{2}(d_i\eta_i - 1) + \hat{y}^T\mathbf{X}\beta - \tfrac{1}{2}\beta^T\mathbf{X}^T\hat{D}\mathbf{X}\beta + \tfrac{1}{2}\hat{u}\hat{\Sigma}^{-1}\hat{u},$$

and we introduce the following notation:

$$\begin{aligned} \hat{\Sigma} &= (\Sigma_0^{-1} + \mathbf{Z}^T D\mathbf{Z})^{-1} \\ \hat{u} &= \hat{\Sigma}\mathbf{Z}^T(y - \tfrac{1}{2}) \\ \hat{D} &= D - D\mathbf{Z}\hat{\Sigma}\mathbf{Z}^T D \\ \hat{y} &= (I - D\mathbf{Z}\hat{\Sigma}\mathbf{Z}^T)(y - \tfrac{1}{2}). \end{aligned}$$

**Introducing the fully-factorized variational approximation.** Similar to the linear regression case, the fully-factorized approximating distribution yields an analytic expression for the lower bound to the marginal log-likelihood:

$$\begin{aligned} &\tfrac{1}{2}\log|\Sigma_0| + \log\Pr(y\,|\,\mathbf{X},\mathbf{Z},\theta) \\ &\quad\geq \tfrac{1}{2}\log|\hat{\Sigma}| + \tfrac{1}{2}\hat{u}^T\hat{\Sigma}^{-1}\hat{u} + \sum_{i=1}^n \log\sigma(\eta_i) + \tfrac{\eta_i}{2}(d_i\eta_i - 1) + \hat{y}^T\mathbf{X}r - \tfrac{1}{2}r^T\mathbf{X}^T\hat{D}\mathbf{X}r \\ &\qquad - \frac{1}{2}\sum_{i=1}^p(\mathbf{X}^T\hat{D}\mathbf{X})_{ii}\mathrm{Var}[\beta_i] + \sum_{i=1}^p \frac{\alpha_i}{2}\left[1 + \log\left(\frac{s_i^2}{\sigma_a^2}\right) - \frac{s_i^2 + \mu_i^2}{\sigma_a^2}\right] \\ &\qquad - \sum_{i=1}^p \alpha_i\log\left(\frac{\alpha_i}{\pi_i}\right) - \sum_{i=1}^p(1-\alpha_i)\log\left(\frac{1-\alpha_i}{1-\pi_i}\right). \end{aligned} \tag{19}$$

As before, $\mathrm{Var}[\beta_i]$ is the variance of $\beta_i$ with respect to the approximating distribution, and $r$ is a column vector with entries $r_i = \alpha_i\mu_i$.

Finding the best fully-factorized distribution amounts to adjusting the free parameters $\theta$ to make the lower bound as tight as possible. The co-ordinate ascent updates for the free parameters are derived by taking partial derivatives of the lower bound, setting these partial derivatives to zero, and solving for $\theta$. This yields the following updates:

$$\mu_i = s_i^2\big((\mathbf{X}^T\hat{y})_i - \textstyle\sum_{j\neq i}(\mathbf{X}^T\hat{D}\mathbf{X})_{ij}\alpha_j\mu_j\big) \tag{20}$$

$$s_i^2 = \big((\mathbf{X}^T\hat{D}\mathbf{X})_{ii} + 1/\sigma_a^2\big)^{-1} \tag{21}$$

$$\frac{\alpha_i}{1-\alpha_i} = \frac{\pi_i}{1-\pi_i} \times \frac{s_i}{\sigma_a} \times e^{\mu_i^2/(2s_i^2)}. \tag{22}$$

The co-ordinate ascent algorithm consists of repeatedly applying these updates until a stationary point is reached.

As in the linear regression case, we derive an approximate EM algorithm to fit the prior variance parameter $\sigma_a^2$. (Recall, $\sigma^2$ is not needed for logistic regression.) The M-step update for $\sigma_a^2$ is identical to Equation 14 after setting $\sigma^2 = 1$.

**Adjusting the linear approximation to the logistic regression likelihood.** In the fourth and final part, we explain how we adjust the parameters $\eta = (\eta_1,\ldots,\eta_n)$ so that the lower bound on the marginal likelihood is as tight as possible. The algorithm is derived interpreting the situation within an EM framework: in the E-step, we compute expectations (the mean and covariance of $\beta$); and in the M-step, we maximize the expected value of the lower bound to the log-likelihood.

We begin by considering the simpler case when we have a single set of variables $X$. Afterward, we substitute to introduce the additional variables $Z$. Taking partial derivatives of $E[g(\beta;\eta)]$ with respect to the variational parameters, we obtain

$$\frac{\partial E[f(\beta;\theta)]}{\partial \eta_i} = \frac{d_i'}{2}(\eta_i^2 - (x_i^T\mu)^2 - x_i^T\Sigma x_i),$$

where $x_i$ is the $i$th row of $\mathbf{X}$, and $\mu$ and $\Sigma$ here are posterior mean and covariance of $\beta$ computed in the E-step. The typical approach is to set the partial derivatives to zero and solve for $\eta$. At first glance, this does not appear to be possible. But a couple of observations will yield a closed-form solution: first, the slope $d$ is symmetric in $\eta$, so we only need to worry about the positive quadrant; second, for $\eta > 0$, $d$ is strictly monotonic as a function of $\eta$, so $d'$ is never zero. Therefore, we can solve for the fixed point:

$$\eta_i^2 = (x_i^T\mu)^2 + x_i^T\Sigma x_i. \tag{23}$$

To derive the M-step update for the fully-factorized variational approximation, after analytically integrating out the coefficients $u$, we need to replace $\mu$ and $\Sigma$ by the correct mean and covariance of $\binom{u}{\beta}$ under the variational approximation. The means and variances of the coefficients $\beta$ are easily obtained from the variational approximation. The remaining means and covariances in Equation 23 are

$$\begin{aligned}
E[u] &= \hat{\Sigma}\mathbf{Z}^T(y - \tfrac{1}{2} - D\mathbf{X}r) \\
\mathrm{Cov}[u] &= \hat{\Sigma} + \hat{\Sigma}\mathbf{Z}^T D\mathbf{X}\mathrm{Cov}[\beta]\mathbf{X}^T D\mathbf{Z}\hat{\Sigma} \\
\mathrm{Cov}[u,\beta] &= -\hat{\Sigma}\mathbf{Z}^T D\mathbf{X}\mathrm{Cov}[\beta].
\end{aligned}$$

Therefore, the final M-step update for $\eta$ is

$$\eta_i^2 = \left(z_i^T E[u] + \textstyle\sum_{j=1}^p x_{ij}E[\beta_j]\right)^2 + z_i^T\mathrm{Cov}[u]z_i + \sum_{j=1}^p x_{ij}^2\mathrm{Var}[\beta_j] + 2z_i^T\mathrm{Cov}[u,\beta]x_i, \tag{24}$$

in which $z_i$ is the $i$th row of $\mathbf{Z}$.

## Affiliation:

Peter Carbonetto
Research Computing Center
University of Chicago
Chicago, Illinois, USA 60637
E-mail: pcarbo@uchicago.edu
URL: http://github.com/pcarbo

*Journal of Statistical Software* http://www.jstatsoft.org/
published by the Foundation for Open Access Statistics http://www.foastat.org/
MMMMMM YYYY, Volume VV, Issue II *Submitted:* yyyy-mm-dd
doi:10.18637/jss.v000.i00 *Accepted:* yyyy-mm-dd