A simple new approach to variable selection in regression, with application to genetic fine-mapping *

Gao Wang and Abhishek Sarkar and Peter Carbonetto and Matthew Stephens

e-mail: gaow@uchicago.edu
aksarkar@uchicago.edu
pcarbo@uchicago.edu
mstephens@uchicago.edu

Abstract: We introduce a simple new approach to variable selection in linear regression, and to quantifying uncertainty in selected variables. The approach is based on a new model – the "Sum of Single Effects" (SuSiE) model – which comes from writing the sparse vector of regression coefficients as a sum of "single-effect" vectors, each with one non-zero element. We also introduce a corresponding new fitting procedure – Iterative Bayesian Stepwise Selection (IBSS) – which is a Bayesian analogue of traditional stepwise selection methods. IBSS shares the computational simplicity and speed of traditional stepwise methods, but instead of selecting a single variable at each step, IBSS computes a *distribution* on variables that captures uncertainty in which variable to select. We show that the IBSS algorithm computes a variational approximation to the posterior distribution under the SuSiE model. Further, this approximate posterior distribution naturally leads to a convenient, novel, way to summarize uncertainty in variable selection, and provides a Credible Set for each selected variable. Our methods are particularly well suited to settings where variables are highly correlated and true effects are very sparse, both of which are characteristics of genetic fine-mapping applications. We demonstrate through numerical experiments that our methods outperform existing methods for this task, and illustrate the methods by fine-mapping genetic variants that influence alternative splicing in human cell-lines. We also discuss both the potential and the challenges for applying these methods to generic variable selection problems.

MSC 2010 subject classifications: Primary 62J05; secondary 62F15.
Keywords and phrases: linear regression, variable selection, sparse, variational, genetic fine-mapping.

1. Introduction

The need to identify, or "select", relevant variables in regression models arises in a diverse range of applications, and has spurred the development of a correspondingly diverse range of methods (e.g., see O'Hara and Sillanpää, 2009; Fan and Lv, 2010; Desboulets, 2018, for reviews). However, variable selection is a

*This work was supported by NIH grant HG002585 and by a grant from the Gordon and Betty Moore Foundation

complex problem, and so despite considerable work there remain important issues that existing methods do not fully address. Here we develop a new approach to this problem that has several attractive features: it is simple, computationally scaleable, and it provides new, more effective, ways to capture uncertainty in which variables should be selected. Our new approach is particularly helpful in situations involving highly correlated variables, where it may be impossible to confidently select any *individual* variable, but it may nonetheless be possible to confidently draw useful conclusions such as "either variable A or B is relevant". More generally it may be possible to confidently identify "Credible Sets" of (correlated) variables, that each, with high probability, contain a relevant variable. Our new approach can quickly, simply and reliably identify such sets, as well as prioritize the variables within each set.

Our work, although potentially of broad interest, is particularly motivated by genetic fine-mapping studies (e.g Veyrieras et al., 2008; Maller et al., 2012; Spain and Barrett, 2015; Huang et al., 2017; Schaid et al., 2018), which aim to identify which genetic variants influence some trait of interest (e.g. LDL cholesterol in blood, gene expression in cells). Genetic fine-mapping can be helpfully framed as a variable selection problem, by building a regression model with the trait as the outcome and genetic variants as predictor variables (Sillanpää and Bhattacharjee, 2005). Performing variable selection in this model identifies variants that may causally affect the trait, and this – rather than prediction accuracy – is the main goal here. Fine-mapping is an example of a variable selection problem that often involves highly correlated variables: neighboring genetic variants are often highly correlated, a phenomenon called *linkage disequilibrium* (Ott, 1999).

Our approach builds on previous work on Bayesian variable selection regression (BVSR) (Mitchell and Beauchamp, 1988; George and McCulloch, 1997), which is already commonly used for genetic fine-mapping and related applications (e.g. Meuwissen et al., 2001; Sillanpää and Bhattacharjee, 2005; Servin and Stephens, 2007; Guan and Stephens, 2011; Bottolo et al., 2011; Maller et al., 2012; Carbonetto and Stephens, 2012; Hormozdiari et al., 2014; Chen et al., 2015; Wallace et al., 2015; Wen et al., 2016; Lee et al., 2018). BVSR has several appealing features compared with other approaches to variable selection. In particular, in principle, BVSR can assess uncertainty about which variables to select, even in the presence of strong correlations among variables. However, applying BVSR in practice remains difficult for two reasons. First, BVSR is computationally challenging, often requiring implementation of sophisticated Markov chain Monte Carlo or stochastic search algorithms (Bottolo and Richardson, 2010; Bottolo et al., 2011; Guan and Stephens, 2011; Wallace et al., 2015; Benner et al., 2016; Wen et al., 2016; Lee et al., 2018). Second, and perhaps more important, the output of existing methods for fitting BVSR is typically a complex posterior distribution, or a sample from it, which can be difficult to distill into easily-interpretable conclusions.

Our new approach addresses these limitations of BVSR through several innovations. First we introduce a new formulation of BVSR, which we call the Sum of Single Effects (*SuSiE*) model. This model, while similar to existing BVSR models, has a different structure that naturally leads to a very simple, intuitive

and fast fitting procedure – Iterative Bayesian Stepwise Selection (IBSS) – which is a Bayesian analogue of traditional stepwise seletion methods. We show that IBSS can be viewed as computing a variational approximation to the posterior distribution under the *SuSiE* model. Unlike previous variational approaches to sparse regression (Logsdon et al., 2010; Carbonetto and Stephens, 2012), this new approach deals well with correlated variables. Furthermore, the approximate posterior leads immediately to Credible Sets of variables that are designed to be as small as possible while still each capturing a relevant variable. Arguably this is exactly the kind of posterior summary that one would like to obtain from MCMC-based or stochastic search BVSR methods, but doing so would require non-trivial post-processing of their output. In contrast, our method provides this posterior summary *directly*, and at a fraction of the computational effort.

The structure of the paper is as follows. In Section 2 we provide further motivation for our work, and brief background on BVSR. Section 3 describes the new *SuSiE* model and fitting procedure. In Section 4 we use simulations, designed to mimic realistic genetic fine-mapping studies, to demonstrate the effectiveness of our approach compared with existing BVSR methods. Section 5 illustrates our methods on fine-mapping of genetic variants affecting splicing, and Section 6 briefly highlights both the promise and the limitations of our methods for other applications using change-point problems. We end with Discussion highlighting avenues for further work.

2. Background

2.1. A motivating toy example

Suppose the relationship between $\boldsymbol{y}$ (an n-vector) and variables $\boldsymbol{X} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_p)$, an $n \times p$ matrix, is modeled as a multiple regression:

$$\begin{aligned} \boldsymbol{y} &= \boldsymbol{X}\boldsymbol{b} + \boldsymbol{e} \\ \boldsymbol{e} &\sim N(0, \sigma^2 I_n), \end{aligned} \tag{2.1}$$

in which $\boldsymbol{b}$ is a p-vector of regression coefficients, $\boldsymbol{e}$ is an n-vector of error terms, $\sigma^2 > 0$ is the residual variance, and I_n is the $n \times n$ identity matrix. For brevity, we will refer to variables j with non-zero effects ($b_j \neq 0$) as "effect variables". We assume that exactly two variables are effect variables – variables 1 and 4, say – and that each of these two effect variables are each completely correlated with another non-effect variable, say $\boldsymbol{x}_1 = \boldsymbol{x}_2$ and $\boldsymbol{x}_3 = \boldsymbol{x}_4$. We further suppose that no other pairs of variables are correlated.

In this situation, given sufficient data, it should be possible to conclude that there are (at least) two effect variables. However, because the effect variables are completely correlated with other variables, it will be impossible to confidently select the correct variables, even when n is very large. The best we can hope to infer is that

$$(b_1 \neq 0 \text{ or } b_2 \neq 0) \text{ and } (b_3 \neq 0 \text{ or } b_4 \neq 0). \tag{2.2}$$

Our goal, in short, is to provide methods that directly produce this kind of inferential statement. Although this example is simplistic, it mimics the kind of structure that occurs in, for example, genetic fine-mapping applications, where it often happens that an association can be narrowed down to a small set of highly correlated genetic variants, and it is desired to provide a quantitative summary about which genetic variants are, based on the data, the plausible effect variables.

Most existing approaches to sparse regression do not provide statements like (2.2). For example, sparse regression methods based on penalized likelihood (e.g., Lasso; Tibshirani, 1996) would, in our example, select one of the four equivalent *configurations* (combinations of variables) – $\{1,3\}$, $\{1,4\}$, $\{2,3\}$ or $\{2,4\}$ – and give no indication that other configurations are equally plausible. Attempting to control error rates of false discoveries at the level of individual variables using methods such as stability selection (Meinshausen and Bühlmann, 2010) or the knockoff filter (Barber and Candès, 2015) will result in no discoveries, since no individual variable can be confidently declared an effect variable. One potential solution would be to first cluster the variables into groups of highly correlated variables and then perform some kind of "group selection" (Huang et al., 2012) or hierarchical testing (Meinshausen, 2008; Mandozzi and Bühlmann, 2016). However, although this might work in our stylized example, in general this approach involves *ad hoc* decisions about which variables to cluster together – an unattractive feature we seek to avoid.

In principle, Bayesian approaches (BVSR; see Introduction for references) can solve this problem. These approaches introduce a prior distribution on the coefficients $\boldsymbol{b}$ to capture the notion that $\boldsymbol{b}$ is sparse, then compute (approximately) a posterior distribution that assesses relative support for each configuration. In our example, the posterior distribution would typically have equal mass (≈ 0.25) on the four equivalent configurations. This posterior distribution contains exactly the information necessary to infer (2.2). However, even in this simple example, translating the posterior distribution to the simple statement (2.2) requires some effort, and in more complex settings such translations become highly non-trivial. Practical implementations of BVSR typically summarize the posterior distribution by the marginal posterior inclusion probability (PIP) of each variable,

$$\mathrm{PIP}_j \coloneqq \Pr(b_j \neq 0 \,|\, \boldsymbol{X}, \boldsymbol{y}). \tag{2.3}$$

In our example, they would report $\mathrm{PIP}_1 = \mathrm{PIP}_2 = \mathrm{PIP}_3 = \mathrm{PIP}_4 \approx 0.5$. While not inaccurate, these marginal PIPs nonetheless fail to convey the information necessary to infer (2.2).

2.2. Credible Sets

To define our main goal more formally, we introduce the concept of a *Credible Set* (CS) of variables:

Definition 1. *In the context of a multiple regression model, we define* ***a level-ρ Credible Set*** *to be a subset of variables that has probability $\geqslant \rho$ of containing*

*at least one effect variable (*i.e.*, a variable with non-zero regression coefficient). Equivalently, the probability that all variables in the Credible Set have zero regression coefficients is no more than* $1 - \rho$.

Our use of the term "Credible Set" here indicates that we have in mind a Bayesian inference approach, in which the probability statements in the definition are statements about uncertainty in the regression coefficients with respect to the available data and modelling assumptions. One could analogously define a "Confidence Set" by interpreting the probability statements as referring to the set, considered random.

Although the term "Credible Set" has been used in fine-mapping applications before, most previous uses either assumed there was a single effect variable (Maller et al., 2012), or defined a CS as a set that contains *all* effect variables (Hormozdiari et al., 2014), which is a very different definition (and, we argue, both less informative and less attainable). Our definition here is closer to the "signal clusters" from Lee et al. (2018). It is also related to the idea of "minimal true detection" in Mandozzi and Bühlmann (2016).

With Definition 1, our primary aim can be restated: we wish to report as many CSs as the data support, each as few variables as possible. For example, to infer (2.2) we would like to report two CSs, $\{1, 2\}$ and $\{3, 4\}$. As a secondary goal, we would also like to prioritize the variables within each CS, assigning each a probability that reflects the strength of the evidence for that variable being an effect variable. Our methods achieve both these goals.

2.3. The single effect regression model

The building block for our approach is the "single effect regression" (SER) model, which we define as a multiple regression model in which *exactly one* of the p explanatory variables has a non-zero regression coefficient. This idea was introduced in Servin and Stephens (2007) to fine-map genetic associations, and consequently adopted and extended by others, such as Veyrieras et al. (2008) and Pickrell (2014). Although of very narrow applicability, the SER model is trivial to fit. Furthermore, when its assumptions hold the SER provides exactly the inferences we desire, including CSs. For example, if we simplify our motivating example (Section 2.1) to have a single effect variable – variable 1, for example – then the SER model would, given sufficient data, infer a 95% CS containing both of the correlated variables, 1 and 2, with PIPs of approximately 0.5 each. This CS tells us that we can be confident that one of the two variables has a non-zero coefficient, but we do not know which one.

Specifically, we consider the following SER model, with hyperparameters σ^2 (the residual variance), σ_0^2 (the prior variance of the non-zero effect) and $\boldsymbol{\pi} = (\pi_1, \ldots, \pi_p)$ (a p-vector of prior inclusion probabilities, with π_j giving the prior

probability that variable j is the effect variable):

$$\boldsymbol{y} = \boldsymbol{X}\boldsymbol{b} + \boldsymbol{e} \tag{2.4}$$

$$\boldsymbol{e} \sim N(0, \sigma^2 I_n) \tag{2.5}$$

$$\boldsymbol{b} = b\boldsymbol{\gamma} \tag{2.6}$$

$$\boldsymbol{\gamma} \sim \text{Mult}(1, \boldsymbol{\pi}) \tag{2.7}$$

$$b \sim N(0, \sigma_0^2). \tag{2.8}$$

Here, $\boldsymbol{y}$ is the n-vector of response data; $\boldsymbol{X}$ is an $n \times p$ matrix; $\boldsymbol{b}$ is a p-vector of regression coefficients; $\boldsymbol{e}$ is an n-vector of independent error terms; and $\text{Mult}(m, \boldsymbol{\pi})$ denotes the multinomial distribution on class counts obtained when m samples are drawn with class probabilities given by $\boldsymbol{\pi}$. (We assume that $\boldsymbol{y}$ and the columns of $\boldsymbol{X}$ have been centered to have mean zero, which avoids the need for an intercept term; see Chipman et al. 2001.)

Under the SER model (2.4–2.8), the effect vector $\boldsymbol{b}$ has exactly one non-zero element (equals to b), so we refer to $\boldsymbol{b}$ as a "single effect vector". The element of $\boldsymbol{b}$ that is non-zero is determined by the binary vector $\boldsymbol{\gamma}$, which also has exactly one non-zero entry. The probability vector $\boldsymbol{\pi}$ determines the prior probability distribution of which of the p variables is the effect variable. In the simplest case, $\boldsymbol{\pi} = (1/p, \ldots, 1/p)$; we assume this uniform prior here but SER model requires only that $\boldsymbol{\pi}$ is fixed and known (so in fine-mapping one could incorporate different priors based on genetic annotations; e.g., Veyrieras et al., 2008). To lighten notation, we henceforth make conditioning on $\boldsymbol{\pi}$ implicit.

Posterior under SER model

Given σ^2 and σ_0^2, the posterior distribution on $\boldsymbol{b} = \boldsymbol{\gamma} b$ is easily computed:

$$\boldsymbol{\gamma} \,|\, \boldsymbol{X}, \boldsymbol{y}, \sigma^2, \sigma_0^2 \sim \text{Mult}(1, \boldsymbol{\alpha}) \tag{2.9}$$

$$b \,|\, \boldsymbol{X}, \boldsymbol{y}, \sigma^2, \sigma_0^2, \gamma_j = 1 \sim N(\mu_{1j}, \sigma_{1j}^2), \tag{2.10}$$

where $\boldsymbol{\alpha} = (\alpha_1, \cdots, \alpha_p)$ is the vector of PIPs, $\alpha_j \coloneqq \Pr(\gamma_j = 1 \,|\, \boldsymbol{X}, \boldsymbol{y}, \sigma^2, \sigma_0^2)$, and μ_{1j}, σ_{1j}^2 are the posterior mean and variance of b given $\gamma_j = 1$. Calculating these quantities simply involves performing the p univariate regressions of $\boldsymbol{y}$ on $\boldsymbol{x}_j$, for $j = 1, \ldots, p$ as detailed in Appendix A. From $\boldsymbol{\alpha}$, it is also simple to compute a level-ρ CS (Definition 1), $CS(\boldsymbol{\alpha}; \rho)$; as described in Maller et al. (2012). In brief, this involves sorting variables by decreasing α_j, then including variables in the CS until their cumulative probability exceeds ρ (Appendix A).

For later convenience we introduce *SER* as shorthand for the function that returns the posterior distribution for $\boldsymbol{b}$ under the SER model. Since this posterior distribution is uniquely determined by the values of $\boldsymbol{\alpha}$, $\boldsymbol{\mu}_1$ and $\boldsymbol{\sigma}_1$ in (2.9-2.10), we can define *SER* as

$$SER(\boldsymbol{y}, \boldsymbol{X}; \sigma^2, \sigma_0^2) \coloneqq (\boldsymbol{\alpha}, \boldsymbol{\mu}_1, \boldsymbol{\sigma}_1^2), \tag{2.11}$$

where $\boldsymbol{\mu}_1 \coloneqq (\mu_{11}, \ldots, \mu_{1p})$ and $\boldsymbol{\sigma}_1^2 \coloneqq (\sigma_{11}^2, \ldots, \sigma_{1p}^2)$.

Empirical Bayes for SER model

Although most previous treatments of the SER model assume σ_0^2 and σ^2 to be fixed and known, we note here the possibility of estimating σ_0^2 and/or σ^2 by maximum-likelihood before computing the posterior distribution of $\boldsymbol{b}$. This is effectively an "Empirical Bayes" approach. The likelihood for σ_0^2 and σ^2 under the SER,

$$L(\sigma_0^2, \sigma^2) := p(\boldsymbol{y} \mid \boldsymbol{X}, \sigma_0^2, \sigma^2), \tag{2.12}$$

is available in closed form (Appendix A), and can be maximized over one or both parameters numerically.

3. The Sum of Single Effects Regression model

We now introduce a new approach to variable selection in multiple regression. Our approach is motivated by the observation that the SER model provides simple inference if there is indeed exactly one effect variable; it is thus desirable to extend the SER to allow for multiple variables. The conventional approach to doing this in BVSR is to introduce a prior on $\boldsymbol{b}$ that allows for multiple non-zero entries (e.g., using the "spike and slab" prior of Mitchell and Beauchamp, 1988). However, this approach no longer enjoys the convenient analytical properties and intuitive interpretation of the posterior distribution under the SER model: posterior distributions become difficult to compute accurately, and CSs even harder.

Here we introduce a different approach, which better preserves the nice features of the SER model. The key idea is straightforward to describe: introduce multiple single-effect vectors, $\boldsymbol{b}_1, \ldots, \boldsymbol{b}_L$, and construct the overall effect vector $\boldsymbol{b}$ as the sum of these single effects. We call this the "SUm of SIngle Effects" (*SuSiE*) regression model:

$$\boldsymbol{y} = \boldsymbol{X}\boldsymbol{b} + \boldsymbol{e} \tag{3.1}$$

$$\boldsymbol{e} \sim N(0, \sigma^2 I_n) \tag{3.2}$$

$$\boldsymbol{b} = \sum_{l=1}^{L} \boldsymbol{b}_l \tag{3.3}$$

$$\boldsymbol{b}_l = \boldsymbol{\gamma}_l b_l \tag{3.4}$$

$$\boldsymbol{\gamma}_l \sim \mathrm{Mult}(1, \boldsymbol{\pi}) \tag{3.5}$$

$$b_l \sim N(0, \sigma_{0l}^2). \tag{3.6}$$

For generality, we have allowed the variance of each effect, σ_{0l}^2, to vary among the components $l = 1, \ldots, L$. The special case in which $L = 1$ recovers the SER model. For simplicity, we initially assume σ^2 and $\boldsymbol{\sigma_0^2} = (\sigma_{01}^2, \ldots, \sigma_{0L}^2)$ are given, and defer estimation of these hyperparameters to Section 3.3.

Note that if $L \ll p$ then the *SuSiE* model is approximately equivalent to a standard BVSR model in which L randomly chosen variables have non-zero

coefficients. The only difference is that with some (small) probability some of the $\boldsymbol{b}_l$ in the *SuSiE* model may have the same non-zero co-ordinates, and so the number of non-zero elements in $\boldsymbol{b}$ has some (small) probability to be less than L. Thus at most L variables have non-zero coefficients in this model. We discuss choice of L in Section 3.5.

Although the *SuSiE* model is approximately equivalent to a standard BVSR model, its novel structure has two major advantages. First, it leads to a simple, iterative and deterministic algorithm for computing approximate posterior distributions. Second, it yields a simple way to calculate the CSs. In essense, because each $\boldsymbol{b}_l$ captures only one effect, the posterior distribution on each γ_l can be used to compute a CS that has a high probability of containing an effect variable. The remainder of this section details both these advantages, and other issues that may arise in fitting the model.

3.1. Fitting SuSiE: Iterative Bayesian stepwise selection

The key to *SuSiE* model fitting procedure is that, given $\boldsymbol{b}_1, \ldots, \boldsymbol{b}_{L-1}$, estimating $\boldsymbol{b}_L$ involves fitting the simpler SER model, which is analytically tractable. This immediately suggests an iterative algorithm that uses the SER model to estimate each $\boldsymbol{b}_l$ in turn, given estimates of the other $\boldsymbol{b}_{l'}$ ($l' \neq l$). Algorithm 1 details such an algorithm, which is both simple and computationally scalable (computational complexity $\mathcal{O}(npL)$ per iteration).

Algorithm 1 Iterative Bayesian stepwise selection

Require: data $\boldsymbol{y}$ and variable matrix $\boldsymbol{X}$.
Require: values for the hyperparameters $\sigma^2, \boldsymbol{\sigma}_0^{\boldsymbol{2}}$ and number of effects L.
Require: a function $SER(\boldsymbol{y}, \boldsymbol{X}; \sigma^2, \sigma_0^2) \rightarrow (\boldsymbol{\alpha}, \boldsymbol{\mu}_1, \boldsymbol{\sigma}_1)$ that computes the posterior distribution for $\boldsymbol{b}$ under the Single Effect Regression model (2.11). [Note: the values of $\boldsymbol{\sigma}_1$ are not strictly required here, but are required for extensions of this algorithm that also estimate σ^2; see Appendix B.]

```
Initialize b̄_l = 0 for l = 1, ..., L.                ▷ other initialization strategies are possible
repeat
    for l in 1, ..., L do
        r_l ← y − Σ_{l'≠l} X b̄_{l'}                   ▷ compute residuals
        (α_l, μ_{1l}, σ_{1l}) ← SER(r_l, X; σ², σ²_{0l})   ▷ fit SER model to residuals
        b̄_l ← α_l ∘ μ_{1l}   ▷ compute posterior mean for b_l; ∘ is element-wise multiplication
until converged
return α_1, μ_{11}, σ_{11}, ..., α_L, μ_{1L}, σ_{1L}.
```

We call Algorithm 1 "Iterative Bayesian Stepwise Selection" (IBSS) because it can be viewed as a Bayesian version of stepwise selection approaches. For example, we can compare it with an approach referred to as "Forward-Stagewise" (FS) selection in Hastie et al. (2009) Section 3.3.3 (although subsequent literature often uses this term slightly differently). In brief, FS first selects the single "best" variable among p candidates by comparing the results of the p univariate regressions. It then computes the residuals from the univariate regression on this selected variable, and selects the next "best" variable by comparing the results

of univariate regression of the residuals on each variable. This process continues, selecting one variable each iteration, until some stopping criteria is reached.

IBSS is similar to FS, but instead of selecting a single "best" variable at each step, it computes a *distribution* on which variable to select, by fitting the Bayesian SER model (Step 5). Similar to FS, this distribution is based on the results of the p univariate regressions, and so IBSS has the same computational complexity as FS ($\mathcal{O}(np)$ per selection). However, by computing a distribution on variables – rather than choosing a single best variable – IBSS captures uncertainty about which variable should be selected at each step. This uncertainty is taken into account when computing residuals (Step 4) by using a *model-averaged* (posterior mean) estimate for the regression coefficients. In IBSS we incorporate an iterative procedure, whereby early selections are re-evaluated in light of the later selections (as in "backfitting"; Friedman and Stuetzle, 1981). The final output of IBSS is L distributions on variables (parameterized by $(\boldsymbol{\alpha}_l, \boldsymbol{\mu}_{1l}, \boldsymbol{\sigma}_{1l})$; $l = 1, \ldots, L$), in place of the L selected variables output by FS. Each distribution is easily summarized by, for example, a 95% CS for each selection.

To illustrate, consider our motivating example (Section 2.1) with $\boldsymbol{x}_1 = \boldsymbol{x}_2$, $\boldsymbol{x}_3 = \boldsymbol{x}_4$, and with variables 1 and 4 having non-zero effects. Suppose for simplicity that the effect of variable 1 is substantially larger than the effect of variable 4. Then FS would first select either variable 1 or 2 (which one being arbitrary), and then select variable 3 or 4 (again, which one being arbitrary). In contrast, given enough data, the first step of IBSS would select both variables 1 and 2 (with equal weight, ≈ 0.5 each, and small weights on other variables). The second step of IBSS would similarly select variables 3 and 4 (again with equal weight, ≈ 0.5 each). Summarizing these results would yield two CSs, $\{1, 2\}$ and $\{3, 4\}$, and the inference (2.2) falls into our lap. This simple example is intended only to sharpen intuition; later numerical experiments demonstrate that IBSS works well in realistic settings.

3.2. IBSS computes a variational approximation to the SuSiE posterior distribution

We now provide a more formal justification for the IBSS algorithm. Specifically, we show that it is a coordinate ascent algorithm for optimizing a *variational approximation* to the posterior distribution for $\boldsymbol{b}_1, \ldots, \boldsymbol{b}_L$ under the *SuSiE* model (3.1)-(3.6). This result also leads to a natural extension of the algorithm to estimate the hyperparameters $\sigma^2, \boldsymbol{\sigma}_0^{\boldsymbol{2}}$.

The idea behind variational approximation methods for Bayesian models (e.g. Blei et al., 2017) is to find an approximation $q(\boldsymbol{b}_1, \ldots, \boldsymbol{b}_L)$ to the posterior distribution $p_{\text{post}} := p(\boldsymbol{b}_1, \ldots, \boldsymbol{b}_L | \boldsymbol{y})$, by minimizing the Kullback–Leibler (KL) divergence from q to p_{post}, $D_{\text{KL}}(q, p_{\text{post}})$, subject to constraints on q that make the problem tractable. Although $D_{\text{KL}}(q, p_{\text{post}})$ is hard to compute, it can be written as:

$$D_{\text{KL}}(q, p_{\text{post}}) = \log p(\boldsymbol{y} | \sigma^2, \boldsymbol{\sigma}_0^{\boldsymbol{2}}) - F(q; \sigma^2, \boldsymbol{\sigma}_0^{\boldsymbol{2}}) \tag{3.7}$$

where F is an easier-to-compute function known as the "evidence lower bound"

(ELBO). (Note: F depends on the data $\boldsymbol{y}, \boldsymbol{X}$, but we suppress this dependence to lighten notation.) Because $\log p(\boldsymbol{y}|\sigma^2, \boldsymbol{\sigma_0^2})$ does not depend on q, minimizing D_{KL} over q is equivalent to maximizing F; and since F is easier to compute this is how the problem is usually framed. See Appendix B.1 for further details.

Here we seek an approximate posterior, q, that factorizes:

$$q(\boldsymbol{b}_1, \ldots, \boldsymbol{b}_L) = \prod_{l=1}^{L} q_l(\boldsymbol{b}_l). \tag{3.8}$$

Under this approximation $\boldsymbol{b}_1, \ldots, \boldsymbol{b}_L$ are independent *a posteriori*. We make no assumptions on the form of q_l, and in particular we do *not* assume that q_l factorizes over the p elements of $\boldsymbol{b}_l$. This is a crucial difference from previous variational approaches for standard multiple regression models (e.g. Logsdon et al., 2010; Carbonetto and Stephens, 2012), and it means that q_l can capture the strong dependencies among the elements of $\boldsymbol{b}_l$ that are induced by the assumption that exactly one element of $\boldsymbol{b}_l$ is non-zero. Intuitively each q_l captures one effect variable, and provides inferences of the form "we need one of variables $\{A, B, C\}$ but we cannot tell which". Similarly, the approximation (3.8) provides inferences of the form "we need one of variables $\{A, B, C\}$ and one of variables $\{D, E, F, G\}$, and ...".

Under (3.8), finding the optimal q can now be written as:

$$\text{maximize}_{q_1,\ldots,q_L} F(q_1, \ldots, q_L; \sigma^2, \boldsymbol{\sigma_0^2}). \tag{3.9}$$

Although jointly optimizing F over $(q_1, \ldots, q_L)$ is not straightforward, it turns out to be very easy to optimize over a single q_l (given $q_{l'}$ for $l' \neq l$), by fitting an SER model, as formalized in the following proposition.

Proposition 1.

$$\arg\max_{q_l} F(q_1, \ldots, q_L; \sigma^2, \boldsymbol{\sigma_0^2}) = SER(\boldsymbol{r}_l, \boldsymbol{X}; \sigma^2, \sigma_{0l}^2) \tag{3.10}$$

where $\boldsymbol{r}_l$ denotes the residuals obtained by removing the estimated effects other than l:

$$\boldsymbol{r}_l := \boldsymbol{y} - \boldsymbol{X} \sum_{l' \neq l} \bar{\boldsymbol{b}}_{l'} \tag{3.11}$$

where $\bar{\boldsymbol{b}}_{l'}$ denotes the expectation of the distribution $q_{l'}$.

For intuition in this proposition, recall that computing the posterior distribution for $\boldsymbol{b}_l$ under the *SuSiE* model if the other effects $\boldsymbol{b}_{l'}, l' \neq l$ were *known* reduces to fitting a SER to residuals $\boldsymbol{y} - \boldsymbol{X} \sum_{l' \neq l} \boldsymbol{b}_{l'}$. Now consider computing an (approximate) posterior distribution for $\boldsymbol{b}_l$ when $\boldsymbol{b}_{l'}$ are not known, but we have approximations $q_{l'}$ to their posterior distributions. This is, essentially, the problem of finding $\arg\max_{q_l} F(q_1, \ldots, q_L)$. Proposition 1 says that we can solve this using a similar procedure as for known $\boldsymbol{b}_{l'}$, but replacing each $\boldsymbol{b}_{l'}$ with its (approximate) posterior mean $\bar{\boldsymbol{b}}_{l'}$. The proof is given in Appendix B (Proposition 2).

The following corollary is an immediate consequence of Proposition 1:

Corollary 1. *Algorithm 1 is a coordinate ascent algorithm for maximizing the ELBO F, and therefore for minimizing the KL divergence $D_{KL}(q, p_{post})$.*

Proof. Step 5 of Algorithm 1 is simply computing the right hand side of equation (3.10). Thus, by Proposition 1, it is a coordinate ascent step for optimizing the lth coordinate of $F(q_1, \ldots, q_L; \sigma^2, \boldsymbol{\sigma}_0^{\mathbf{2}})$ (the distribution q_l being determined by the parameters $\boldsymbol{\alpha}_l, \boldsymbol{\mu}_{1l}, \boldsymbol{\sigma}_{1l}$). □

3.3. Estimating $\boldsymbol{\sigma}^{\mathbf{2}}, \boldsymbol{\sigma}_{\mathbf{0}}^{\mathbf{2}}$

Algorithm 1 can be extended to estimate the hyperparameters σ^2 and $\boldsymbol{\sigma}_0^{\mathbf{2}}$, by adding steps to optimize $F(q_1, \ldots, q_L; \sigma^2, \boldsymbol{\sigma}_0^{\mathbf{2}})$ over σ^2 and/or $\boldsymbol{\sigma}_0^{\mathbf{2}}$. Estimating hyperparameters by maximizing the ELBO F is a commonly-used strategy in variational inference, and often performs well in practice (e.g. Carbonetto and Stephens, 2012).

Optimizing F over σ^2 involves computing the expected residual sum of squares under the variational approximation, which is straightforward; see Appendix B for details.

Optimizing F over $\boldsymbol{\sigma}_0^{\mathbf{2}} = (\sigma_{01}^2, \ldots, \sigma_{0L}^2)$ can be achieved by modifying the step that computes the posterior distribution for $\boldsymbol{b}_l$ under the SER model (Step 5) to first estimate the hyperparameter σ_{0l}^2 in the SER model by maximum likelihood. That is, by optimizing (2.12) over σ_0^2, keeping σ^2 fixed. This is a one-dimensional optimization which is easily performed numerically (we used the R function `uniroot`).

Algorithm 4 in Appendix B extends IBSS to include both these steps.

3.4. Posterior inference: posterior inclusion probabilities and Credible Sets

Algorithm 1 provides an approximation to the posterior distribution on $\boldsymbol{b}$ under the *SuSiE* model, parameterized by $(\boldsymbol{\alpha}_1, \boldsymbol{\mu}_{11}, \boldsymbol{\sigma}_{11}), \ldots, (\boldsymbol{\alpha}_L, \boldsymbol{\mu}_{1L}, \boldsymbol{\sigma}_{1L})$. From these results it is straightforward to compute approximations to various posterior quantities of interest, including PIPs and CSs.

3.4.1. Posterior inclusion probabilities

Under the *SuSiE* model the effect b_j is zero if and only if $b_{lj} = 0$ for all l. Under the variational approximation the b_{lj} are independent across l, and so:

$$\mathrm{PIP}_j := \Pr(b_j \neq 0 | \boldsymbol{y}, \boldsymbol{X}) \approx 1 - \prod_{l \in \mathcal{L}} (1 - \boldsymbol{\alpha}_{l_j}). \tag{3.12}$$

Here, $\mathcal{L} = \{l : \sigma_{0l}^2 > 0\}$, to account for the edge case where some $\sigma_{0l}^2 = 0$ (which can happen when $\boldsymbol{\sigma}_0^{\mathbf{2}}$ is estimated as in Section 3.3).

3.4.2. Credible Sets

Simply computing the sets $CS(\boldsymbol{\alpha}_l; \rho)$ (A.17) for $l = 1, \ldots, L$ immediately yields L CSs that satisfy Definition 1 under the variational approximation to the posterior.

If L exceeds the number of detectable effects in the data then in practice it turns out that many of the L CSs are large, often containing the majority of variables. The intuition is that once all the detectable signals have been accounted for, the IBSS algorithm becomes very uncertain about which variable to include at each step, and so the distributions $\boldsymbol{\alpha}$ become very diffuse. CSs that contain very many uncorrelated variables are of essentially no inferential value – whether or not they actually contain a effect variable – and so in practice it makes sense to ignore them. To automate this process, in this paper we discard CSs with purity < 0.5, where we define purity as the smallest absolute correlation among all pairs of variables within the CS. (To reduce computation for CSs containing >100 variables we sampled 100 variables at random to compute purity.) The purity threshold 0.5 was chosen primarily for comparability with Lee et al. (2018) who use a similar threshold in a related context. Although any choice of threshold is somewhat arbitrary, in practice we observed that most CSs are either very pure (> 0.95) or very impure (< 0.05), with intermediate cases being rare (Figure S2), and so results are robust to this choice of threshold.

3.5. Choice of L

It may seem that choice of suitable L would be crucial. However, in practice key inferences are robust to overstating L; for example in our simulations later the true L is 1-5, but we obtain good results with $L = 10$. This is because, when L is too large, the method becomes very uncertain about where to place the additional (non-existent) effects; consequently it distributes them broadly among many variables, and so they have little impact on key inferences. For example, setting L too large inflates the PIPs of many variables just very slightly, and leads to some low-purity CSs that are filtered out (see Section 3.4.2).

Although inferences are generally robust to overstating L, we also note that the Empirical Bayes version of our method, which estimates $\boldsymbol{\sigma}_0^{\mathbf{2}}$, also effectively estimates the number of effects: when L is greater than the number of signals in the data, the maximum likelihood estimate of σ_{0l}^2 is often 0 for many l, which forces $b_l = 0$. This is connected to "Automatic relevance determination" (Neal, 1996).

4. Numerical Comparisons

We use simulations to assess our methods and compare with standard BVSR methods. Our simulations are designed to mimic genetic fine-mapping studies, in particular fine-mapping of expression quantitative trait loci (eQTLs) – eQTLs are genetic variants associated with gene expression.

In genetic fine-mapping, $\boldsymbol{X}$ is a matrix of genotype data, in which each row corresponds to an individual, and each column corresponds to a genetic variant, typically a single nucleotide polymorphism (SNP). In our simulations, we used the real human genotype data from the $n = 574$ genotype samples collected as part of the Genotype-Tissue Expression (GTEx) project (GTEx Consortium, 2017). To simulate fine-mapping of *cis* effects on gene expression, we randomly select 150 genes from the $> 20{,}000$ genes on chromosomes 1–22, then take $\boldsymbol{X}$ to be the genotypes for genetic variants nearby the transcribed region of the selected gene. For a given gene, between $p = 1{,}000$ and $p = 12{,}000$ SNPs are included in the fine-mapping analysis; for more details on how SNPs are selected, see Appendix C.

To assess accuracy of *SuSiE* inference by comparing estimates against "ground-truth", we generate synthetic gene expression data $\boldsymbol{y}$ under the multiple regression model (2.1), with various assumptions on the effects $\boldsymbol{b}$. We specify our assumptions about the simulated effects $\boldsymbol{b}$ using two parameters: S, the number of effect variables; and ϕ, the proportion of variance in $\boldsymbol{y}$ explained by $\boldsymbol{X}$ ("PVE" for short).

We consider two sets of simulations. In the first set, each data set has exactly $p = 1,000$ SNPs. We simulate data sets under all combinations of $S \in \{1, \ldots, 5\}$ and $\phi \in \{0.05, 0.1, 0.2, 0.4\}$. These settings were chosen to span typical values for eQTL studies. Given choices of S and ϕ, we take the following steps to simulate gene expression data:

(a) Sample the indices $\mathcal{S}$ of the S effect variables uniformly at random from $\{1, \ldots, p\}$.
(b) For each $j \in \mathcal{S}$, draw $b_j \sim N(0, 0.6^2)$, and set $b_j = 0$ for all $j \notin \mathcal{S}$.
(c) Set σ^2 to achieve the desired PVE, ϕ; specifically, we solve for σ^2 in $\phi = \frac{\mathrm{Var}(\boldsymbol{X}\boldsymbol{b})}{\sigma^2 + \mathrm{Var}(\boldsymbol{X}\boldsymbol{b})}$, where $\mathrm{Var}(\,\cdot\,)$ denotes sample variance.
(d) For each $i = 1, \ldots, 574$, draw $y_i \sim N(x_{i1}b_1 + \cdots + x_{ip}b_p, \sigma^2)$.

We simulate two replicates for each gene and each scenario, resulting in a total of $2 \times 150 \times 4 \times 5 = 6{,}000$ simulations.

In the second set of simulations, we generate larger data sets with 3,000 to 12,000 SNPs. To simulate gene expression data, we set $S = 10$ and $\phi = 0.3$. Again, we simulate two replicates for each gene, so in total, we generate an additional $2 \times 150 = 300$ data sets in the second set of simulations.

4.1. Illustrative example

We begin with an example to illustrate that, despite its simplicity, the IBSS algorithm (Algorithm 1) can perform well in a challenging situation. This example is given in Figure 1.

We draw this example from one of our simulations in which the variable most strongly associated with $\boldsymbol{y}$ is not one of the actual effect variables (in this particular example, there are two effect variables). This situation occurs because at least one variable has moderate correlation with both effect variables, and these

effects combine to make its marginal association stronger than the marginal associations of the individual effect variables. Standard forward selection in this case would select the wrong variable in the first step; by contrast, the iterative nature of IBSS allows it to escape this trap. Indeed, in this example IBSS yields two high-purity CSs, each containing one of the effect variables.

Interestingly, in this example the most strongly associated variable does not appear in either CS. This illustrates that multiple regression can sometimes result in very different conclusions compared to a marginal association analysis. An animation showing the iteration-by-iteration progress of the IBSS algorithm can be found at our manuscript resource repository (Wang et al., 2018).

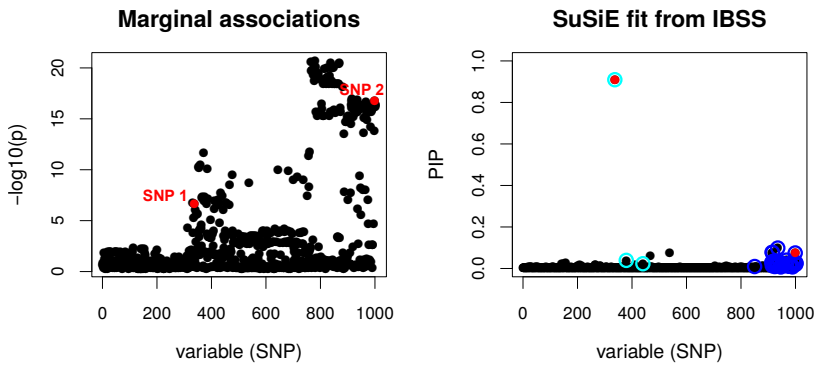


Figure 1: Illustration that IBSS algorithm can deal with a challenging case. Results are from a simulated data with $p = 1,000$ variables (the SNPs), of which two are effect variables (labeled as "SNP 1" and "SNP 2", in red). This example was chosen because the strongest marginal association is with a non-effect variable (at position 780 on the x-axis); see the p values in the left-hand panel. Despite its simplicity, the IBSS algorithm converges to a solution in which the two 95% CSs – represented by the light and dark blue open circles in the right-hand panel – each contain a true effect variable. Additionally, neither CS contains the variable that has the strongest marginal association. One CS contains only 3 SNPs, whereas the other CS (in dark blue) contains 37 very highly correlated variables (minimum pairwise absolute correlation of 0.972). In the latter CS, the individual PIPs are small, but the inclusion of the 37 variables in this CS indicates, correctly, high confidence in one effect variable among them.

4.2. Posterior inclusion probabilities

Next we seek to assess the effectiveness of our methods more quantitatively. We focus initially on one of the simpler tasks in BVSR: computing posterior inclusion probabilities (PIPs). Most implementations of BVSR compute PIPs, making it possible to compare results across several implementations. Here we

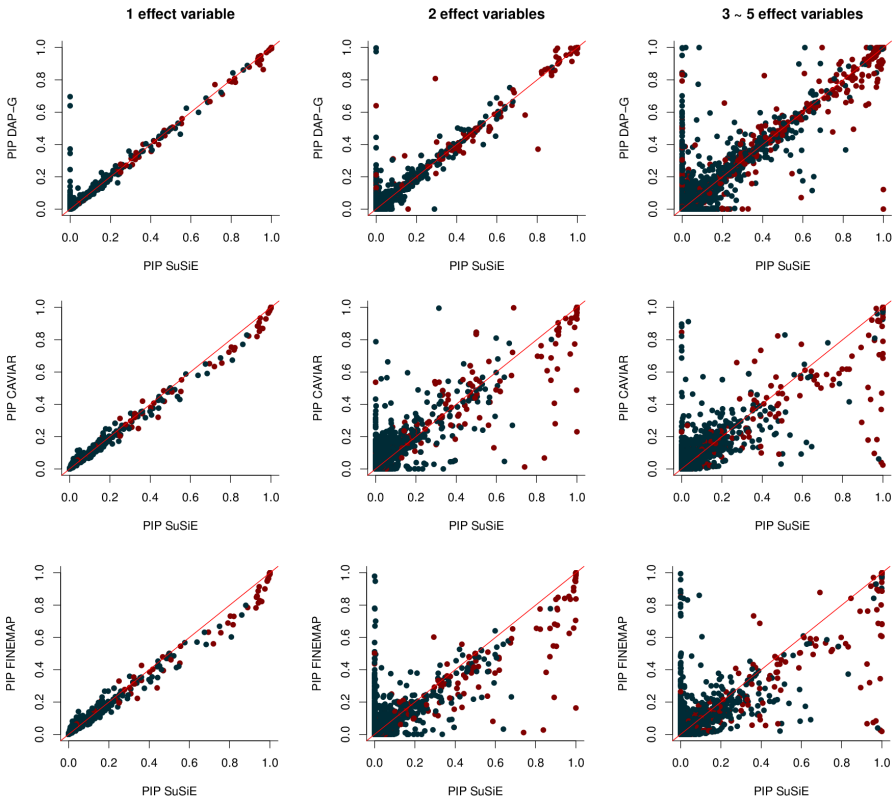
compare our methods (henceforth *SuSiE* for short, implemented in an R package, `susieR`, version 0.4.29) with three other software implementations aimed at genetic fine-mapping applications: CAVIAR (Hormozdiari et al., 2014, version 2.2), FINEMAP (Benner et al., 2016, version 1.1) and DAP-G (Wen et al., 2016; Lee et al., 2018, GitHub commit `ef11b26`). These C++ software packages implement different algorithms to fit similar BVSR models, which differ in details such as priors on effect sizes. CAVIAR exhaustively evaluates all possible combinations of up to L non-zero effects among the p variables, whereas FINEMAP and DAP-G approximate this exhaustive approach by heuristics that target the best combinations. Another difference among methods is that FINEMAP and CAVIAR perform inference using summary statistics computed for each dataset – specifically, the marginal association Z scores and the $p \times p$ correlation matrix for all variables – whereas, as we apply them here, DAP-G and *SuSiE* use the full data. The summary statistic approach can be viewed as approximating inferences from the full data; see Lee et al. (2018) for discussion.

For *SuSiE*, we set $L = 10$ for the first set of simulations, and $L = 20$ for the data sets with the larger numbers of SNPs. We assessed performance when estimating both hyperparameters $\sigma^2, \boldsymbol{\sigma_0^2}$, and when fixing one or both of these. Overall, results from these different strategies were similar. In the main text, we show results obtained when estimating σ^2 and fixing σ_{0l}^2 to $0.1\mathrm{Var}(\boldsymbol{y})$, to be consistent with data applications in Section 5; other results are found in Supplementary Data (Figure S4, Figure S5). Parameter settings for other methods are given in Appendix C. Since CAVIAR and FINEMAP were much more computationally intensive than DAP-G and *SuSiE*, we ran all methods in simulations with $S = 1, 2, 3$, and only ran DAP-G and *SuSiE* in simulations with $S > 3$.

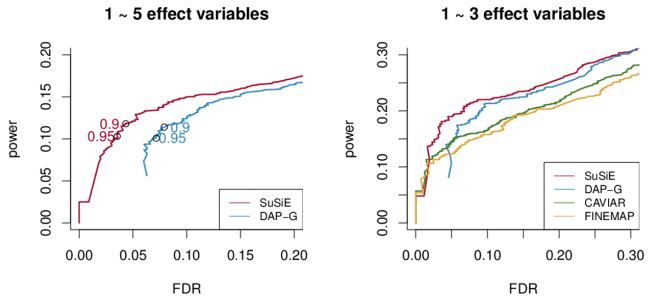
Since these methods differ in their modelling assumptions, we do not expect their PIPs to agree exactly. Nonetheless, we found generally good agreement (Figure 2A). For $S = 1$, the PIPs from all four methods closely agree. For $S > 1$, the PIPs from different methods are also highly correlated; correlations between PIPs from *SuSiE* and other methods vary from 0.94 to 1, and the proportions of PIPs differing by more than 0.1 between methods vary from 0.013% to 0.2%. Visually, this agreement appears less strong because the eye is drawn to the small proportion of points that lie away from the diagonal, while the vast majority of points lie on or near the origin. In addition, all four methods produce reasonably well-calibrated PIPs (Figure S1).

The general agreement of PIPs from four different methods suggests that: (i) all four methods are mostly accurate for computing PIPs for the size of the problems explored in our numerical comparisons; and (ii) the PIPs themselves are typically robust to details of the modelling assumptions. Nonetheless, non-trivial differences in PIPs are clearly visible from Figure 2A. Visual inspection of the results of these simulations suggests that, in many of these cases, *SuSiE* assigns higher PIPs to the true effect variables than other methods, particularly compared to FINEMAP and CAVIAR; for non-effect variables where other methods report high PIPs, *SuSiE* often correctly assigns PIPs close to zero. These observations suggest that the PIPs from *SuSiE* may better distinguish effect variables from non-effect variables. This is confirmed by our analysis

A. Direct comparison of Posterior Inclusion Probability



B. Power vs. False Discovery Rate



of power vs. False Discovery Rate (FDR) for each method, which is obtained by varying the PIP threshold for each method (Figure 2B): the *SuSiE* PIPs always yield comparable or higher power at a given FDR.

Notably, even implemented in R, *SuSiE* computations are much faster than others implemented in C++: in the $S = 3$ simulations, *SuSiE* is roughly 4 times faster than DAP-G, 30 times faster than FINEMAP, and 4,000 times faster than CAVIAR on average (Table 1).

In summary, the results suggest that *SuSiE* produces PIPs that are as or more reliable than existing methods, and does so at a fraction of the computational cost.

Table 1: Runtimes on data sets simulated with $S = 3$ (all times are in seconds)

Method	Avg.	Min.	Max.
SuSiE	0.64	0.34	2.28
DAP-G	2.87	2.23	8.87
FINEMAP	23.01	10.99	48.16
CAVIAR	2907.51	2637.34	3018.52

4.3. Credible Sets

A key feature of *SuSiE* is that it yields multiple Credible Sets (CSs), *each aimed at capturing an effect variable* (Definition 1). The only other BVSR method that attempts something similar – as far as we are aware – is DAP-G, which outputs "signal clusters" defined by heuristic rules (Lee et al., 2018). Although Lee et al. (2018) do not refer to their signal clusters as CSs, and do not give a formal definition of signal cluster, the signal clusters have a similar goal to our CSs, and so for brevity we henceforth refer to them as CSs.

Figure 2 *(preceding page)*: Evaluation of posterior inclusion probabilities (PIPs). Scatterplots in **Panel A** compare PIPs computed in *SuSiE* against PIPs computed using other methods (DAP-G, CAVIAR, FINEMAP). Points drawn in red represent true effect variables; point in black represent variables of no effect. Each scatterplot combines results from many simulations. **Panel B** summarizes these same results as a plot of power vs. FDR. These curves are obtained by varying the PIP threshold for each method. The open circles in the left-hand plot highlight results at PIP thresholds of 0.9 and 0.95). Here, FDR $\coloneqq \frac{\mathrm{FP}}{\mathrm{TP+FP}}$ (also known as the "false discovery proportion"), and power $\coloneqq \frac{\mathrm{TP}}{\mathrm{TP+FN}}$, where FP, TP, FN and TN denote the number of False Positives, True Positives, False Negatives and True Negatives, respectively. (This plot is the same as a *precision-recall curve* after reversing the x-axis, because precision $= \frac{\mathrm{TP}}{\mathrm{TP+FP}} = 1 - \mathrm{FDR}$, and recall = power.)

We compared the level 95% CSs produced by *SuSiE* and DAP-G in several ways. First we assessed their empirical (frequentist) coverage levels, i.e., proportion of CSs that contain an effect variable. Since our CSs are Bayesian Credible Sets, they are not designed or guaranteed to have frequentist coverage 0.95 (Fraser, 2011). Indeed, coverage will inevitably depend on simulation scenario. For example, in completely null simulations ($\boldsymbol{b} = 0$) *every* CS would necessarily contain no effect variable, and so coverage would be 0. Nonetheless, one might hope that under reasonable simulations that include effect variables the Bayesian CSs would have coverage near the nominal levels. And indeed, we confirmed this was the case: in these simulations CSs from both methods typically had coverage slightly below 0.95, but usually > 0.9 (Figure 3; see Figure S3 for additional results).

Having established that the methods produce CSs with similar coverage, we compared them by three other criteria: (i) power (overall proportion of simulated effect variables included in a CS); (ii) average size (median number of variables included in CS) and (iii) purity (here measured as average squared correlation of variables in CS since this is output by DAP-G). By all three metrics the CSs from *SuSiE* are consistently better: higher power, smaller size and higher purity (Figure 3).

Although the way that we construct CSs in *SuSiE* does not require that they be disjoint, we note that in practice here CSs rarely overlapped (after filtering out low purity CSs; Section 3.4.2). Indeed, across all simulations there was only one instance of a pair of overlapping CSs.

5. Application to splice QTL fine-mapping

5.1. Genome-wide identification of splice QTL in human cell lines

To illustrate *SuSiE* on a real fine-mapping problem we analyzed data from Li et al. (2016) aimed at detecting genetic variants (SNPs) that influence splicing (known as "splice QTLs", sQTLs). These authors quantified alternative splicing by estimating, at each intron in each sample, a ratio capturing how often the intron is used relative to other introns in the same "cluster" (roughly, gene). The data involve 77,345 intron ratios measured on lymphoblastoid cell lines from 87 Yoruban individuals, together with genotypes of these individuals. Following Li et al. (2016) we pre-process the intron ratios by regressing out the first 3 principle components of the matrix of intron ratios, which aims to control for unmeasured confounders (Leek and Storey, 2007). For each intron ratio we test for its association with SNPs within 100kb of the intron, which is on average ~ 600 SNPs. In other words, we run *SuSiE* on 77,345 data sets with $n = 87$ and $p \approx 600$.

To specify the prior variance σ_{0l}^2 we first estimated typical effect sizes from the data on all introns. Specifically we performed simple (SNP-by-SNP) regression analysis at every intron, and estimated the PVE of the top (strongest associated) SNP. The mean PVE of the top SNP across all introns was 0.096, and so we

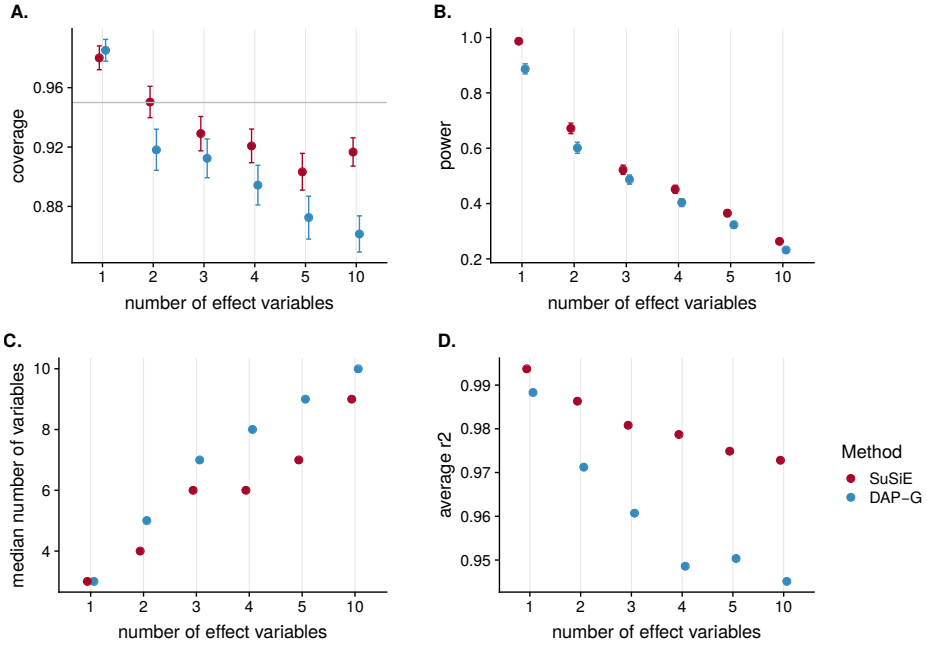


Figure 3: Comparison of 95% credible sets (CS) from *SuSiE* and DAP-G. Panels show A) coverage, B) power, C) median size and D) average squared correlation of the variables in each CS. Scenarios with 1-5 effect variables each involved $p = 1,000$ variables. The Scenario with 10 effect variables involved $p = 3,000 - 12,000$ variables (the entire candidate region in cis-eQTL association analysis of GTEx data).

applied *SuSiE* with $\sigma_{0l}^2 = 0.096\text{Var}(\boldsymbol{y})$ (with the columns of $\boldsymbol{X}$ standardized to have unit variance).

We then applied *SuSiE* to fine-map sQTLs at all 77,345 introns. After filtering for purity, this yielded a total of 2,652 CSs (level 0.95) which were spread across 2,496 intron units. These numbers are broadly in line with the original study, which reported 2,893 significant introns at 10% FDR. Of the 2,652 CSs identified, 457 contain exactly one SNP, and these represent strong candidates for being actual causal variants that affect splicing. Another 239 CSs contain exactly two SNPs. The median size of CS was 7 and the median purity was 0.94.

The vast majority of intron units with any CS had only one CS (2,357 of 2,496). Thus, at most introns *SuSiE* could reliably identify (at most) one sQTL. Of the remainder, 129 introns yielded two CSs, 5 introns yielded three CSs, 3 introns yielded four CSs and 2 introns yielded five CSs. This represents a total of 156 (129+10+9+8) additional ("secondary") signals that would be missed in conventional analyses that report only one signal per intron. Although these data show relatively few secondary signals, this is a relatively small study ($n = 87$); it is likely that in larger studies the ability of *SuSiE* (and other fine-mapping

methods) to detect secondary signals will be more important.

5.2. Functional enrichment of association signals

Although in these real data we do not know the true causal SNPs, we can provide indirect evidence that both the primary and secondary signals identified here are enriched for real signals using functional enrichment analysis. To perform this analysis we labelled one CS at each intron the "primary" CS, and we chose the CS with highest purity at each intron as the primary CS; remaining CSs at each intron (if any) were labelled "secondary" CSs. We then tested both primary and secondary CSs for enrichment of SNPs with various biological annotations, by comparing SNPs inside these CSs (with PIP>0.2) against random control SNPs outside CSs.

We used the same annotations in our enrichment analysis as Li et al. (2016). These were: LCL-specific histone marks (H3K27ac, H3K27me3, H3K36me3, H3K4me1, H3K4me2, H3K4me3, H3K79me2, H3K9ac, H3K9me3, H4K20me1), DNase I hypersensitive sites, transcriptional repressor CTCF binding sites, RNA polymerase II (PolII) binding sites and extended splice sites (defined as 5bp up/down-stream of intron start site and 15bp up/down-stream of intron end site), and intron and coding annotations.

Figure 4 shows the enrichments in both primary and secondary CSs, for annotations that were significant at p-value $< 10^{-4}$ in the primary signals (Fisher's exact test, two-sided). The strongest enrichment in both primary and secondary signals was for extended splice sites (odds ratio ≈ 5 in primary signals), which is reassuring given that these results are for splice QTLs. Other significantly enriched annotations in primary signals include PolII binding, several histone marks, and coding regions. The only annotation showing a significant depletion was introns. Results for secondary signals were qualitatively similar to those for primary, though all enrichments are less significant due to the much smaller numbers of secondary CSs.

6. An example beyond fine-mapping: change point detection

Although our methods were initially motivated by genetic fine-mapping applications, they are also applicable to other sparse regression applications. Here we briefly illustrate this by applying *SuSiE* to an example that is very different from fine-mapping: change point detection. This application also provides a simple example where the IBSS algorithm produces a poor fit – due to getting stuck in a local optimum – which is something we seldom observed in fine-mapping simulations. We believe that examples where algorithms fail are just as important as examples where they succeed – perhaps more so – and that this example could help motivate future methods development and improvements.

In brief, we consider a simple change point model:

$$y_t = \mu_t + e_t \quad t = 1, \ldots, T \tag{6.1}$$

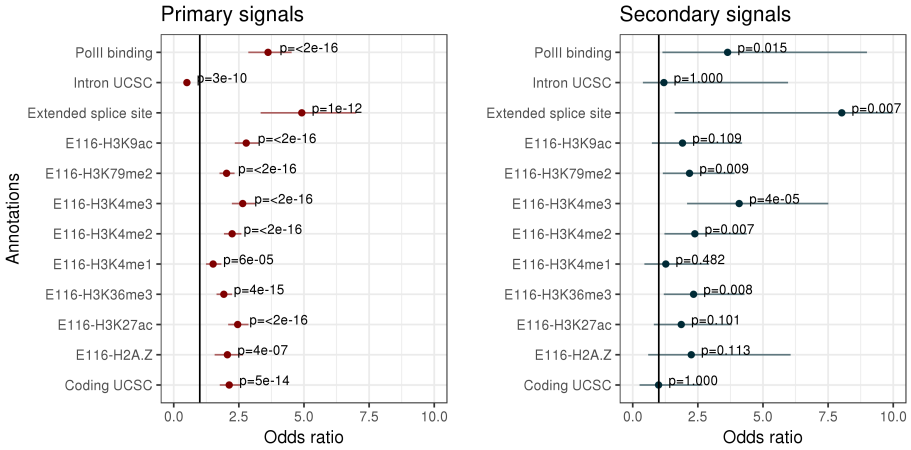


Figure 4: Results of enrichment analysis for splice QTLs. The plot shows the estimated odds ratio, $\pm$ 2 standard errors, for each annotation, obtained by comparing the annotations of SNPs inside primary/secondary CSs against random control SNPs outside CSs (see text for definitions of primary and secondary). The p-values are from two-sided Fisher's exact test.

where t indexes location in one-dimensional space (or time), the errors e_t are independent and identically distributed $N(0, \sigma^2)$, and the mean vector $\boldsymbol{\mu} := (\mu_1, \ldots, \mu_T)$ is assumed to be piecewise constant. Indices, t, at which $\boldsymbol{\mu}$ changes ($\mu_t \neq \mu_{t+1}$) are then called "change-points".

The idea that change points are rare can be captured by formulating this model as a sparse multiple regression (2.1), where $\boldsymbol{X}$ has $T-1$ columns, the tth column being a step function with step at t ($x_{st} = 0$ for $s \leqslant t$ and 1 for $s > t$). The tth element of $\boldsymbol{b}$ then determines the change in the mean at position t, $\mu_{t+1} - \mu_t$, and so the non-zero regression coefficients in this multiple regression model correspond to change points in $\boldsymbol{\mu}$.

Note that the design matrix $\boldsymbol{X}$ has a very different structure here from in fine-mapping applications. In particular, the correlation matrix of the columns of $\boldsymbol{X}$ has its largest elements near the diagonal, and gradually decays moving away from the diagonal – very different from the "blocky" correlation structure that typically occurs in genetic fine-mapping. (A side note on computation: due to the special structure of this $\boldsymbol{X}$, *SuSiE* computations can be made $O(TL)$ rather than $O(T^2L)$ which would be achieved by a naive implementation; for example the vector $\boldsymbol{X}^T\boldsymbol{y}$ is simply the cumulative sums of the elements of the reverse of $\boldsymbol{y}$, which can be computed in linear complexity.)

Change point detection has a wide range of potential applications, including, for example, segmentation of genomes into regions with different numbers of copies of the genome. Software packages in R that can be used for detecting change points include `changepoint` (Killick and Eckley, 2014), `DNAcopy` (Seshan and Olshen, 2018; Olshen et al., 2004), `bcp` (Erdman and Emerson, 2007)

and `genlasso` (Tibshirani, 2014; Arnold and Tibshirani, 2016); see Killick and Eckley (2014) for a longer list. Of these, only `bcp`, which implements a Bayesian method, quantifies uncertainty in estimated change point locations, and `bcp` provides only (marginal) PIPs, not CSs for change point locations. Therefore the ability of *SuSiE* to provide such CSs is unusual, and perhaps even unique, among existing methods for this problem.

To illustrate the potential of *SuSiE* for change point estimation we applied it to a simple simulated example from the `DNAcopy` package. Figure 5 shows both the *SuSiE* and `DNAcopy` results. The two methods provide similar estimates for the change point locations, but *SuSiE* also provides a 95% CS for each change point. In this case every true change point is contained in a reported CS, and every CS contains a true change point. This is true even though our fit assumed $L = 10$ change points and the truth is only 7 change points: the additional CSs were filtered out here because they contain very uncorrelated variables. (Actually *SuSiE* reported 8 CSs after filtering, two of them overlapping and containing the same true change point. As reported in Section 4, in fine-mapping applications we found such overlapping CSs very rarely.)

To highlight an example where IBSS can converge to a poor local optima, consider the simple simulated example in Figure 5, which consists of two change points in quick succession that approximately cancel each other out (so the mean before and after the change points are equal). We designed this example specifically to illustrate a limitation of IBSS: here introducing any single change point (to the null model of no change point) makes the fit worse, and one really needs to introduce both change points at the same time to improve the fit, which IBSS is not set up to do. Consequently, when run from a null initialization, IBSS finds no change points (and reports no CSs).

We emphasize that this result represents a limitation of the IBSS algorithm for optimizing the objective function, and not a limitation of either the *SuSiE* model or the variational approximation. To demonstrate this we re-ran the IBSS algorithm, initialized from a solution that contains the two true change points. This yields a fit with two CSs, containing the two correct change points, and a higher value of the objective function than the original fit (-148.2 vs -181.8). Improved fitting algorithms – or more careful initialization of IBSS – could therefore address this problem.

7. Discussion

We presented a new model (*SuSiE*) and algorithm (IBSS) which together provide a simple new approach to variable selection in regression. Compared with existing methods, the main benefits of our approach are its computational efficiency, and its ability to provide CSs summarizing uncertainty in which variables should be selected. Our numerical comparisons demonstrate that for genetic fine-mapping our methods outperform existing methods at a fraction of the computational cost.

Although our methods apply generally to variable selection in linear regression, further work may be required to improve performance in difficult settings.

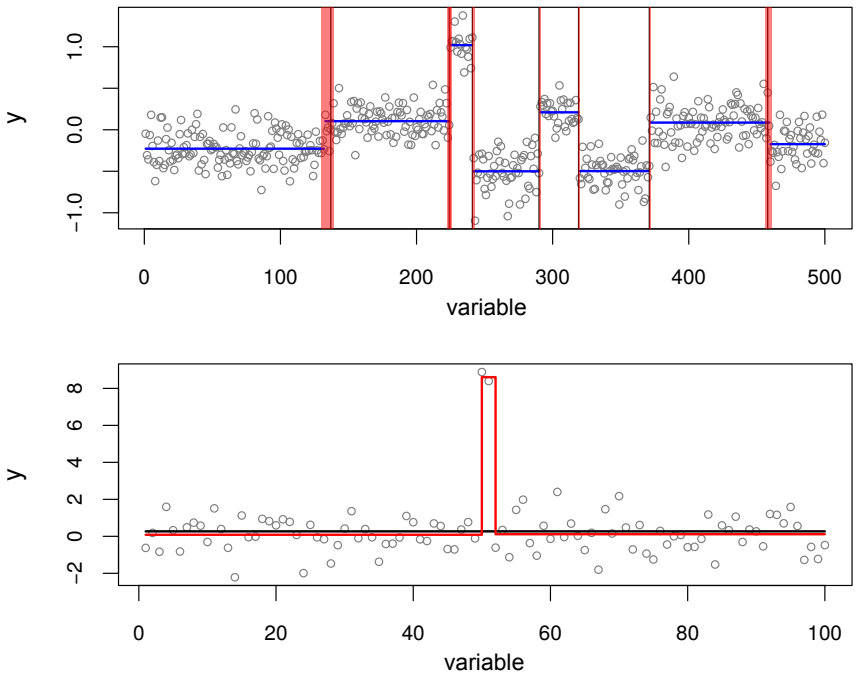


Figure 5: Illustration of *SuSiE* applied to two simple changepoint problems. Top panel shows a simple simulated example with seven true change points (vertical black lines). The blue horizontal lines show the mean function inferred by `DNAcopy::segment`. The inference is reasonably accurate, but provides no indication of uncertainty in change point locations. The red vertical strips show the 95% CSs for change point locations inferred by *SuSiE*. Each CS contains a true change point. **Bottom panel** shows a simple simulated example with two change points in quick succession, designed to show how the IBSS algorithm used to fit *SuSiE* can converge to a local optimum. The two lines shows the fit from initializing IBSS from the null model with no change points (black), and the true model with two change points (red). The red line is much closer to the truth and attains a higher value of the objective function (-148.2 vs -181.8)

In particular, while the IBSS algorithm worked well in our fine-mapping experiments, for change-point problems we showed that IBSS may converge to poor local optima. We have also seen convergence problems in experiments with many effect variables (e.g. 200 effect variables out of 1,000). Such problems may be alleviated by better initialization, for example using fits from convex objective functions (e.g. Lasso) or from more sophisticated algorithms for non-convex problems (Bertsimas et al., 2016; Hazimeh and Mazumder, 2018). More ambitiously, one could attempt to develop better algorithms to reliably optimize the *SuSiE* variational objective function in difficult cases. For example, taking smaller steps each iteration, rather than full coordinate ascent, may help.

At its core, the *SuSiE* model is based on adding up simple models (SERs) to create more flexible models (sparse multiple regression). This additive structure is the key to our variational approximations, and indeed our methods apply generally to adding up any simple models for which exact Bayesian calculations are tractable, not only SER models (Appendix B; Algorithm 3). These observations suggest connections with both additive models and boosting (e.g. Friedman et al., 2000; Freund et al., 2017). However, our methods differ from most work on boosting in that each "weak learner" (here, SER model) itself yields a model-averaged predictor. Other differences include our use of back-fitting, the potential to estimate hyper-parameters by maximizing an objective function rather than cross-validation, and the interpretation of our algorithm as a variational approximation to a Bayesian posterior. Although we did not focus on prediction accuracy here, the generally good predictive performance of methods based on model averaging and boosting suggest that *SuSiE* should work well for prediction as well as variable selection.

It would be natural to extend our methods to generalized linear models (glms), particularly logistic regression. In genetic studies with small effects Gaussian models are often adequate to model binary outcomes (e.g. Zhou et al., 2013). However, in other settings this extension may be more important. One strategy would be to directly modify the IBSS algorithm, replacing the SER fitting procedure with a logistic or glm equivalent. This strategy is appealing in its simplicity, although it is not obvious what objective function the resulting algorithm is optimizing.

For genetic fine-mapping it would also be useful to modify our methods to deal with settings where only summary data are available (e.g. the p simple regression results). Many recent fine-mapping methods deal with this (e.g Chen et al., 2015; Benner et al., 2016; Newcombe et al., 2016) and ideas used by these methods can also be applied to *SuSiE*. Indeed our software already includes preliminary implementations for this problem.

Finally, we are particularly interested in extending these methods to select variables simultaneously for multiple outcomes (*multivariate regression*, and *multi-task learning*). In settings where multiple outcomes share the same relevant variables, multivariate analysis can greatly enhance power and precision to identify relevant variables. The computational simplicity of our approach makes it particularly appealing for this complex task, and we are currently pursuing this by combining our methods with those from Urbut et al. (2018).

8. Data and resources

SuSiE is implemented in the R package `susieR` available at https://github.com/stephenslab/susieR. Source code and a website documenting in detail the analysis steps for numerical comparisons and data applications are available at our manuscript resource repository Wang et al. (2018), also available at https://github.com/stephenslab/susie-paper.

Acknowledgements

We thank Kaiqian Zhang and Yuxin Zou for their substantial contributions to the development and testing of the `susieR` package. Computing resources were provided by the University of Chicago Research Computing Center. This work was supported by NIH grant HG002585 and by a grant from the Gordon and Betty Moore Foundation (Grant GBMF #4559).

Appendix A: Details of posterior computations for the SER model

A.1. Bayesian simple linear regression

To provide posterior computations for the SER it helps to start with the even simpler Bayesian simple linear regression model:

$$\boldsymbol{y} = \boldsymbol{x}b + \boldsymbol{e} \tag{A.1}$$

$$\boldsymbol{e} \sim N(0, \sigma^2 I_n) \tag{A.2}$$

$$b \sim N(0, \sigma_0^2). \tag{A.3}$$

Here $\boldsymbol{y}$ is an n-vector of response data (centered to have mean 0), $\boldsymbol{x}$ is an n-vector containing values of a single explanatory variable (similarly centered), $\boldsymbol{e}$ is an n-vector of independent error terms with variance σ^2, b is the scalar regression coefficient to be estimated, and σ_0^2 is the prior variance of b.

Given σ^2, σ_0^2 the Bayesian computations for this model are very simple. They can be conveniently written in terms of the usual least-squares estimate of b, $\hat{b} := (\boldsymbol{x}^T\boldsymbol{x})^{-1}\boldsymbol{x}^T\boldsymbol{y}$, its variance, $s^2 := \sigma^2/(\boldsymbol{x}^T\boldsymbol{x})$, and the corresponding z score $z := \hat{b}/s$. The posterior distribution for b is

$$b|\boldsymbol{y}, \sigma^2, \sigma_0^2 \sim N(\mu_1, \sigma_1^2) \tag{A.4}$$

where

$$\sigma_1^2(\boldsymbol{x}; \sigma^2, \sigma_0^2) := (s^{-2} + \sigma_0^{-2})^{-1}, \tag{A.5}$$

$$\mu_1(\boldsymbol{y}, \boldsymbol{x}; \sigma^2, \sigma_0^2) := (\sigma_1^2/s^2)\hat{b}. \tag{A.6}$$

And the Bayes Factor (BF) for comparing this model with the null model ($b = 0$) is:

$$\mathrm{BF}(\boldsymbol{y}, \boldsymbol{x}; \sigma^2, \sigma_0^2) := \frac{p(\boldsymbol{y}|\boldsymbol{x}, \sigma^2, \sigma_0^2)}{p(\boldsymbol{y}|\boldsymbol{x}; \sigma^2, b = 0)} \tag{A.7}$$

$$= \sqrt{\frac{s^2}{\sigma_0^2 + s^2}} \exp\left(\frac{z^2}{2} \frac{\sigma_0^2}{\sigma_0^2 + s^2}\right). \tag{A.8}$$

(The form of BF matches the "asymptotic BF" of Wakefield (2009), but here – because we consider linear regression and given σ^2 – it is an exact expression and not only asymptotic.)

A.2. The single effect regression model

Under the SER model, given $\sigma^2, \sigma_0^2, \boldsymbol{\pi}$, the posterior distribution on $\boldsymbol{b} = \boldsymbol{\gamma} b$ is as given in the main text:

$$\boldsymbol{\gamma}|\boldsymbol{y}, \sigma^2, \sigma_0^2 \sim \mathrm{Mult}(1, \boldsymbol{\alpha}); \tag{A.9}$$

$$b|\gamma_j = 1, \boldsymbol{y}, \sigma^2, \sigma_0^2 \sim N(\mu_{1j}, \sigma_{1j}^2), \tag{A.10}$$

where $\boldsymbol{\alpha} = (\alpha_1, \cdots, \alpha_p)$ is the vector of PIPs:

$$\alpha_j = \mathrm{BF}(\boldsymbol{y}, \boldsymbol{x}_j; \sigma^2, \sigma_0^2)\pi_j / \sum_{j'=1}^{p} \mathrm{BF}(\boldsymbol{y}, \boldsymbol{x}_{j'}; \sigma^2, \sigma_0^2)\pi_{j'} \tag{A.11}$$

with BF as in (A.8), and where μ_{1j}, σ_{1j}^2 are the posterior mean (A.6) and variance (A.5) from Bayesian simple regression of $\boldsymbol{y}$ on $\boldsymbol{x}_j$:

$$\mu_{1j} = \mu_1(\boldsymbol{y}, \boldsymbol{x}_j; \sigma^2, \sigma_0^2) \tag{A.12}$$

$$\sigma_{1j} = \sigma_1(\boldsymbol{x}_j; \sigma^2, \sigma_0^2). \tag{A.13}$$

Our algorithm requires the first and second moments of this posterior distribution, which are:

$$\mathrm{E}[b_j|\boldsymbol{y}, \sigma^2, \sigma_0^2] = \alpha_j \mu_{1j} \tag{A.14}$$

$$\mathrm{E}[b_j^2|\boldsymbol{y}, \sigma^2, \sigma_0^2] = \alpha_j(\sigma_{1j}^2 + \mu_{1j}^2). \tag{A.15}$$

A.3. Computing Credible Sets

As noted in the main text, under the SER model it is simple to compute a level ρ CS (Definition 1), $CS(\boldsymbol{\alpha}; \rho)$, as described in Maller et al. (2012). For convenience we give the procedure here explicitly.

Given $\boldsymbol{\alpha}$, let $r = (r_1, \ldots, r_p)$ denote the indices of the variables ranked in order of decreasing $\boldsymbol{\alpha}_j$, so $\boldsymbol{\alpha}_{r_1} > \boldsymbol{\alpha}_{r_2} > \cdots > \boldsymbol{\alpha}_{r_p}$, and let S_k denote the cumulative sum of the k largest PIPs:

$$S_k := \sum_{j=1}^{k} \boldsymbol{\alpha}_{r_j}. \tag{A.16}$$

Now take

$$CS(\boldsymbol{\alpha}; \rho) := \{r_1, \ldots, r_{k_0}\} \tag{A.17}$$

where $k_0 = \min\{k : S_k \geqslant \rho\}$. This choice of k_0 ensures that the CS is as small as possible while still satisfying the requirement that it has at least level ρ.

A.4. Empirical Bayes approach

As noted in the main text, it is possible to take an Empirical Bayes approach to estimating the hyperparameters σ^2, σ_0^2. The likelihood is:

$$L_{\text{SER}}(\sigma_0^2, \sigma^2; \boldsymbol{y}) := p(\boldsymbol{y}|\boldsymbol{X}, \sigma_0^2, \sigma^2) = p_0(\boldsymbol{y}|\sigma^2) \sum_{j=1}^{p} \text{BF}(\boldsymbol{y}, \boldsymbol{x}_j; \sigma^2, \sigma_0^2)\pi_j, \tag{A.18}$$

where p_0 denotes the distribution of $\boldsymbol{y}$ under the "null" that $b = 0$ (i.e. $N(0, \sigma^2 I_n)$). The likelihood (A.18) can be maximized over one or both parameters numerically.

Appendix B: Derivation of Variational Algorithms

B.1. Background: Empirical Bayes and Variational Approximation

Here we introduce helpful background and notation before applying the ideas to our specific application.

B.1.1. Empirical Bayes as a single optimization problem

Consider problems of the form:

$$\boldsymbol{y} \sim p(\boldsymbol{y}|\boldsymbol{b}, \theta) \tag{B.1}$$

$$\boldsymbol{b} \sim g(\boldsymbol{b}) \tag{B.2}$$

where $\boldsymbol{y}$ represent observed data, $\boldsymbol{b}$ represent unobserved (latent) variables of interest, $g \in \mathcal{G}$ represents a prior distribution for $\boldsymbol{b}$ (which in the Empirical Bayes paradigm is treated as an unknown to be estimated) and $\theta \in \Theta$ represents parameters to be estimated. This formulation also includes as a special case situations where g is pre-specified rather than estimated, simply by making $\mathcal{G}$ contain a single distribution.

Fitting this model by Empirical Bayes involves the following steps:

1. Obtain estimates $\hat{g}, \hat{\theta}$ for g, θ, by maximizing the log-likelihood:

$$(\hat{g}, \hat{\theta}) := \arg \max_{g \in \mathcal{G}, \theta \in \Theta} l(g, \theta; \boldsymbol{y}) \tag{B.3}$$

   where

$$l(g, \theta; \boldsymbol{y}) := \log \int p(\boldsymbol{y}|\boldsymbol{b}, \theta) g(\boldsymbol{b}) d\boldsymbol{b}. \tag{B.4}$$

2. Compute the posterior distribution for $\boldsymbol{b}$ given these estimates, $\widehat{p_{\text{post}}}(\boldsymbol{b}) := p_{\text{post}}(\boldsymbol{b}; \boldsymbol{y}, \hat{g}, \hat{\theta})$ where

$$p_{\text{post}}(\boldsymbol{b}; \boldsymbol{y}, g, \theta) := p(\boldsymbol{b}|\boldsymbol{y}, g, \theta). \tag{B.5}$$

This two step procedure can be conveniently written as a single optimization problem:

$$(\widehat{p_{\text{post}}}, \hat{g}, \hat{\theta}) = \arg \max_{q, g \in \mathcal{G}, \theta \in \Theta} F(q, g, \theta; \boldsymbol{y}), \tag{B.6}$$

with

$$F(q, g, \theta; \boldsymbol{y}) := l(g, \theta; \boldsymbol{y}) - D_{\text{KL}}(q || p_{\text{post}}(\cdot; \boldsymbol{y}, g, \theta)) \tag{B.7}$$

where

$$D_{\text{KL}}(q||p) := -\int \log \frac{p(\boldsymbol{b})}{q(\boldsymbol{b})} q(\boldsymbol{b}) d\boldsymbol{b} \tag{B.8}$$

is the Kullback–Leibler divergence from q to p and the optimization over q in (B.6) is over *all possible distributions*. The function F (B.7) is often called the "evidence lower bound", or ELBO, because it is a lower bound for the evidence (log-likelihood). (This follows from the fact that KL divergence is non-negative.)

That this single optimization problem (B.6) is equivalent to the usual two-step EB procedure follows from two simple observations:

1. Since the log-likelihood, l, does not depend on q, we have

$$\arg \max_q F(q, g, \theta; \boldsymbol{y}) = \arg \min_q D_{\text{KL}}(q || p_{\text{post}}(\cdot; g, \theta, \boldsymbol{y})) = p_{\text{post}}(\cdot; g, \theta, \boldsymbol{y}). \tag{B.9}$$

2. Since the optimum D_{KL} term over q is 0 for any θ, g we have $\max_q F(q, g, \theta; \boldsymbol{y}) = l(g, \theta; \boldsymbol{y})$, so

$$(\hat{g}, \hat{\theta}) := \arg \max_{g \in \mathcal{G}, \theta \in \Theta} l(g, \theta; \boldsymbol{y}) = \arg \max_{g \in \mathcal{G}, \theta \in \Theta} \max_q F(q, g, \theta; \boldsymbol{y}). \tag{B.10}$$

B.1.2. Variational approximation

The optimization problem (B.6) is often intractable. The idea of variational approximation is to adjust the problem to make it tractable, simply by restricting the optimization over q to $q \in \mathcal{Q}$ where $\mathcal{Q}$ denotes a suitably chosen class of distributions:

$$(\widehat{p_{\text{post}}}, \hat{g}, \hat{\theta}) = \arg \max_{q \in \mathcal{Q}, g \in \mathcal{G}, \theta \in \Theta} F(q, g, \theta; \boldsymbol{y}). \tag{B.11}$$

From the definition of F it follows that optimizing F over $q \in \mathcal{Q}$ (for given g, θ) corresponds to minimizing the KL divergence from q to the posterior distribution, and so can be interpreted as finding the "best" approximation to the posterior distribution for $\boldsymbol{b}$ among distributions in the class $\mathcal{Q}$. And the optimization of F over g, θ can be thought of as replacing the optimization of the log-likelihood with optimization of the ELBO, a lower bound to the log-likelihood.

We refer to solutions of the general problem (B.6), in which q is unrestricted, as "EB solutions". We refer to solutions of the restricted problem (B.11) as "Variational EB (VEB) solutions".

B.1.3. Algebraic form for F

It is helpful to note that, by simple algebraic manipulations, the ELBO F in (B.7) can be written as:

$$F(q, g, \theta; \boldsymbol{y}) = \mathrm{E}_q \left[\log \frac{p(\boldsymbol{y}, \boldsymbol{b} | g, \theta)}{q(\boldsymbol{b})} \right] \tag{B.12}$$

$$= \mathrm{E}_q \left[\log p(\boldsymbol{y} | \boldsymbol{b}, \theta) \right] + \mathrm{E}_q \left[\log \frac{g(\boldsymbol{b})}{q(\boldsymbol{b})} \right]. \tag{B.13}$$

B.2. The additive effects model

We now focus on fitting an additive model, $\mathcal{M}$, that includes the *SuSiE* model as a special case:

$$\boldsymbol{y} = \sum_{l=1}^{L} \boldsymbol{\mu}_l + \boldsymbol{e} \tag{B.14}$$

$$\boldsymbol{\mu}_l \sim g_l \quad \text{independently for } l = 1, \ldots, L \tag{B.15}$$

$$\boldsymbol{e} \sim N(0, \sigma^2 I_n), \tag{B.16}$$

where $\boldsymbol{y} \in R^n, \boldsymbol{\mu}_l \in R^n, \boldsymbol{e} \in R^n$, and I_n denotes the $n \times n$ identity matrix. We let $\mathcal{M}_l$ denote the simpler model that is derived from $\mathcal{M}$ by setting $\boldsymbol{\mu}_{l'} \equiv 0$ for $l' \neq l$ (i.e. $\mathcal{M}_l$ is the model that contains only the lth additive term), and use L_l to denote the marginal likelihood for this simpler model:

$$L_l(\boldsymbol{y}; g_l, \sigma^2) := p(\boldsymbol{y} | \mathcal{M}_l, g_l, \sigma^2). \tag{B.17}$$

The *SuSiE* model corresponds to the special case of $\mathcal{M}$ where $\boldsymbol{\mu}_l := X\boldsymbol{b}_l$ and g_l is the "single effect prior" in (2.6)-(2.8). Further, in this special case each $\mathcal{M}_l$ is a "single effect regression" (SER) model.

The key idea is that we can fit $\mathcal{M}$ by VEB provided we can fit each simpler model $\mathcal{M}_l$ by EB. To expand on this: consider fitting the model $\mathcal{M}$ by

VEB, where the restricted family $\mathcal{Q}$ is the class of distributions on $(\boldsymbol{\mu}_l)_{l=1}^L$ that factorize over l. That is, for any $\boldsymbol{q} \in \mathcal{Q}$,

$$\boldsymbol{q}(\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_L) = \prod_{l=1}^{L} q_l(\boldsymbol{\mu}_l), \tag{B.18}$$

and we can write $\boldsymbol{q} = (q_1, \ldots, q_L)$.

For $\boldsymbol{q} \in \mathcal{Q}$, using expression (B.13), we obtain the following expression for the ELBO F:

$$F(\boldsymbol{q}, \boldsymbol{g}, \sigma^2; \boldsymbol{y}) = -\frac{n}{2}\log(2\pi\sigma^2) - \frac{1}{2\sigma^2}\mathrm{E}_{\boldsymbol{q}}\|\boldsymbol{y} - \sum_l \boldsymbol{\mu}_l\|^2 + \sum_l \mathrm{E}_{q_l}\left[\log\frac{g_l(\boldsymbol{\mu}_l)}{q_l(\boldsymbol{\mu}_l)}\right], \tag{B.19}$$

where $\|\boldsymbol{y}\|^2 := \boldsymbol{y}^T\boldsymbol{y}$ and $\boldsymbol{g}$ denotes the priors $(g_1, \ldots, g_L)$. Note that the second term here is the expected residual sum of squares (ERSS) under $\boldsymbol{q}$, and depends on $\boldsymbol{q}$ only through its first and second moments. Indeed, if we define

$$(\bar{\boldsymbol{\mu}}_l)_i := \mathrm{E}_{q_l}\mu_{li} \tag{B.20}$$

$$(\overline{\boldsymbol{\mu}_l^2})_i := \mathrm{E}_{q_l}[(\mu_{li})^2], \tag{B.21}$$

and $\bar{\boldsymbol{\mu}} := (\bar{\boldsymbol{\mu}}_l)_{l=1}^L, \overline{\boldsymbol{\mu}^2} := (\overline{\boldsymbol{\mu}_l^2})_{l=1}^L$, then

$$\mathrm{ERSS}(\boldsymbol{y}, \bar{\boldsymbol{\mu}}, \overline{\boldsymbol{\mu}^2}) = \|\boldsymbol{y} - \sum_l \bar{\boldsymbol{\mu}}_l\|^2 - \sum_l \bar{\boldsymbol{\mu}}_l^T\bar{\boldsymbol{\mu}}_l + \sum_l\sum_i \overline{\boldsymbol{\mu}_l^2}_i. \tag{B.22}$$

(This expression follows from the definition, and independence across $l = 1, \ldots, L$, by simple algebraic manipulation; see Section B.6).

Fitting $\mathcal{M}$ by VEB involves optimizing F in (B.19) over $\boldsymbol{q}, \boldsymbol{g}, \sigma^2$. Our strategy is to use "coordinate ascent", using steps that optimize over (q_l, g_l) $(l = 1, \ldots, L)$ while keeping other elements of $\boldsymbol{q}, \boldsymbol{g}$ fixed, and with a separate step to optimize over σ^2 given $\boldsymbol{q}, \boldsymbol{g}$. This strategy is summarized in Algorithm 2.

Algorithm 2 Coordinate Ascent for F (outline)

```
repeat
    for l in 1, ..., L do
        (q_l, g_l) ← arg max_{q_l, g_l} F(q, g, σ²; y)        ▷ update q_l, g_l
    σ² ← arg max_{σ²} F(q, g, σ²; y)                          ▷ update σ²
until converged
```

The update for σ^2 in Algorithm 2 is easily obtained by taking partial derivative of (B.19), setting to zero, and solving for σ^2, giving

$$\hat{\sigma}^2 := \frac{\mathrm{ERSS}(\boldsymbol{y}, \bar{\boldsymbol{\mu}}, \overline{\boldsymbol{\mu}^2})}{n}. \tag{B.23}$$

The update for (q_l, g_l) turns out to correspond to finding the EB solution to the simpler model $\mathcal{M}_l$, but with the data $\boldsymbol{y}$ replaced with the expected residuals,

$\bar{\boldsymbol{r}}_l := \boldsymbol{y} - \sum_{l' \neq l} \bar{\boldsymbol{\mu}}_{l'}$. The proof of this is given in the next section (Proposition 2).

Substituting these ideas into Algorithm 2 gives Algorithm 3, which is a generalization of the IBSS algorithm (Algorithm 1) in the main text.

Algorithm 3 Coordinate Ascent for F

```
Initialize μ_l = 0; μ_l^2 = 0 for l = 1, ..., L.          ▷ other initializations are possible
repeat
    for l in 1, ..., L do
        r̄_l ← y − Σ_{l'≠l} μ̄_{l'}          ▷ compute residuals from removing all effects except l
        g_l ← arg max L_l(r̄_l; g_l, σ^2)          ▷ EB update of g_l (optional; omitted if g_l fixed)
        Compute posterior q_l = p(μ_l | M_l, r̄_l, g_l)          ▷ update q_l
        μ̄_l ← E_{q_l}(μ_l)          ▷ first moment
        μ̄_l^2 ← E_{q_l}(μ_l^2)          ▷ second moment; see (B.21)
    σ^2 ← (1/n)ERSS(y, μ̄, μ̄^2) from (B.22)          ▷ update σ^2
until converged
```

B.3. Special case of SuSiE model

The *SuSiE* model corresponds to the special case $\boldsymbol{\mu}_l = \boldsymbol{X}\boldsymbol{b}_l$, in which case $\mathcal{M}_l$ is a single effects regression model. The first and second moments of $\boldsymbol{\mu}_l$, $\bar{\boldsymbol{\mu}}_l$ and $\overline{\boldsymbol{\mu}^2}_l$ are determined by the first and second moments of $\boldsymbol{b}_l$:

$$\mathrm{E}[\mu_{lj}] = \boldsymbol{X}\mathrm{E}[b_{lj}] \tag{B.24}$$

$$\mathrm{E}[\mu_{li}^2] = \mathrm{E}[(\sum_j \boldsymbol{X}_{ij} b_{lj})^2] \tag{B.25}$$

$$= \sum_j \boldsymbol{X}_{ij}^2 \mathrm{E}[b_{lj}^2] \tag{B.26}$$

where the last line comes from the fact that only one element of $\boldsymbol{b}_l$ is nonzero, so the cross terms $b_{lj} b_{lj'} = 0$ for $j \neq j'$. Because of this we can write $\mathrm{ERSS}(\boldsymbol{y}, \bar{\boldsymbol{\mu}}, \overline{\boldsymbol{\mu}^2})$ as a function of the first and second moments of the $\boldsymbol{b}_l$ – say $\mathrm{ERSS}(\boldsymbol{y}, \bar{\boldsymbol{b}}, \overline{\boldsymbol{b}^2})$ – and Algorithm 3 can be implemented by working with the posterior distributions of $\boldsymbol{b}$ instead of $\boldsymbol{\mu}$.

For completeness we give this algorithm, which is the one we implemented in the `susieR` software, explicitly as Algorithm 4. This algorithm is the same as the IBSS algorithm in the main text, but extended to estimate the hyperparameters $\sigma^2, \boldsymbol{\sigma}_0^{\boldsymbol{2}}$.

Algorithm 4 Iterative Bayesian stepwise selection (Extended)

Require: data $\boldsymbol{y}$ and variable matrix $\boldsymbol{X}$.
Require: (initial) values for the hyperparameters $\sigma^2, \boldsymbol{\sigma_0^2}$.
Require: value for number of effects L.
Require: a function $SER(\boldsymbol{y}, \boldsymbol{X}; \sigma^2, \sigma_0^2) \to (\boldsymbol{\alpha}, \boldsymbol{\mu}_1, \boldsymbol{\sigma}_1)$ that computes the posterior distribution for $\boldsymbol{b}$ under the Single Effect Regression model (2.11).

1: Initialize $\bar{\boldsymbol{b}}_l = \overline{\boldsymbol{b}^2}_l = 0$ for $l = 1, \ldots, L$. ▷ other initialization strategies are possible
2: **repeat**
3: **for** l in $1, \ldots, L$ **do**
4: $\boldsymbol{r}_l \leftarrow \boldsymbol{y} - \sum_{l' \neq l} \boldsymbol{X}\bar{\boldsymbol{b}}_{l'}$ ▷ compute residuals
5: $\sigma_{0l}^2 \leftarrow \arg\max_{\sigma_0^2} L_{\mathrm{SER}}(\sigma_0^2, \sigma^2; \boldsymbol{r}_l)$ ▷ optional EB step; L_{SER} given at (A.18)
6: $(\boldsymbol{\alpha}_l, \boldsymbol{\mu}_{1l}, \boldsymbol{\sigma}_{1l}) \leftarrow SER(\boldsymbol{r}_l, \boldsymbol{X}; \sigma^2, \sigma_{0l}^2)$ ▷ fit SER model to residuals
7: $\bar{\boldsymbol{b}}_l \leftarrow \boldsymbol{\alpha}_l \circ \boldsymbol{\mu}_{1l}$ ▷ posterior mean for $\boldsymbol{b}_l$; $\circ$ is element-wise multiplication
8: $\overline{\boldsymbol{b}^2}_l \leftarrow \boldsymbol{\alpha}_l \circ (\boldsymbol{\sigma_{1l}^2} + \boldsymbol{\mu}_{1l}^2)$ ▷ posterior second moment for $\boldsymbol{b}_l$
9: $\sigma^2 \leftarrow (1/n)\mathrm{ERSS}(\boldsymbol{y}, \bar{\boldsymbol{b}}, \overline{\boldsymbol{b}^2})$. ▷ update σ^2
10: **until** converged
11: **return** $\boldsymbol{\alpha}_1, \boldsymbol{\mu}_{11}, \boldsymbol{\sigma}_{11}, \ldots, \boldsymbol{\alpha}_L, \boldsymbol{\mu}_{1L}, \boldsymbol{\sigma}_{1L}$.

We implemented the option Step 5, which is a one-dimensional optimization, using `uniroot` in R to find the point where the derivative of L_{SER} is 0.

B.4. Update for q_l, g_l is EB solution of $\mathcal{M}_l$

In this subsection we establish that Step 3 in Algorithm 2 is accomplished by EB solution of $\mathcal{M}_l$ (Steps 4–5 in Algorithm 3). This result is formalized in the following Proposition, which generalizes Proposition 1 in the main text:

Proposition 2. *Optimizing F in (B.19) over q_l, g_l is achieved by*

$$\arg\max_{q_l, g_l} F(\boldsymbol{q}, \boldsymbol{g}, \sigma^2; \boldsymbol{y}) = \arg\max_{q_l, g_l} F_l(q_l, g_l, \sigma^2; \boldsymbol{y} - \sum_{l' \neq l} \bar{\boldsymbol{\mu}}_{l'}). \tag{B.27}$$

where F_l denotes the ELBO corresponding to model $\mathcal{M}_l$ and $\bar{\boldsymbol{\mu}}_l$ is as in (B.20).

Note that the optimization of F_l over q_l, g_l on the right hand side of (B.27) does not involve restrictions on q_l, and so corresponds precisely to finding the EB solution to M_l (see Section B.1.1).

Proof. Let F_l denote the ELBO for model $\mathcal{M}_l$. Then, from (B.13) we have:

$$F_l(q_l, g_l, \sigma^2; \boldsymbol{y}) = -\frac{n}{2}\log(2\pi\sigma^2) - \frac{1}{2\sigma^2}\mathrm{E}_{q_l}\|\boldsymbol{y} - \boldsymbol{\mu}_l\|^2 + \mathrm{E}_{q_l}\left[\log\frac{g_l(\boldsymbol{\mu}_l)}{q_l(\boldsymbol{\mu}_l)}\right]. \tag{B.28}$$

Further, let $\boldsymbol{\mu}_{-l}$ denote the components of $(\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_L)$ omitting $\boldsymbol{\mu}_l$, and $\boldsymbol{q}_{-l}$ denote the distribution on $\boldsymbol{\mu}_{-l}$ induced by marginalizing $\boldsymbol{q}$ over $\boldsymbol{b}_l$. Finally, let $\boldsymbol{r}_l$ denote the residuals obtained by removing all the effects other than l from $\boldsymbol{y}$, and $\bar{\boldsymbol{r}}_l$ denote its expectation under $\boldsymbol{q}_{-l}$:

$$\boldsymbol{r}_l := \boldsymbol{y} - \sum_{l' \neq l} \boldsymbol{\mu}_{l'}, \tag{B.29}$$

$$\bar{\boldsymbol{r}}_l := \mathrm{E}_{\boldsymbol{q}_{-l}}\left[\boldsymbol{r}_l(\boldsymbol{\mu}_{-l})\right] = \boldsymbol{y} - \sum_{l' \neq l} \bar{\boldsymbol{\mu}}_{l'}. \tag{B.30}$$

Now, separating F in (B.19) into the parts that depend on q_l, g_l, and those that do not (here denoted "const"), we have:

$$F(\boldsymbol{q}, \boldsymbol{g}, \sigma^2; \boldsymbol{y}) = -\frac{1}{2\sigma^2}\mathrm{E}_{\boldsymbol{q}}\left[\left(\boldsymbol{r}_l - \boldsymbol{\mu}_l\right)^T\left(\boldsymbol{r}_l - \boldsymbol{\mu}_l\right)\right] + \mathrm{E}_{q_l}\left[\log\frac{g_l(\boldsymbol{\mu}_l)}{q_l(\boldsymbol{\mu}_l)}\right] + \text{const} \tag{B.31}$$

$$= -\frac{1}{2\sigma^2}\mathrm{E}_{q_l}\left[\left(-2\bar{\boldsymbol{r}}_l^T\boldsymbol{\mu}_l + \boldsymbol{\mu}_l^T\boldsymbol{\mu}_l\right)\right] + \mathrm{E}_{q_l}\left[\log\frac{g_l(\boldsymbol{\mu}_l)}{q_l(\boldsymbol{\mu}_l)}\right] + \text{const} \tag{B.32}$$

$$= F_l(q_l, g_l, \sigma^2; \bar{\boldsymbol{r}}_l) + \text{const}. \tag{B.33}$$

□

B.5. Computing the evidence lower bound

Although not strictly required to implement Algorithm 3, it can also be helpful to compute the objective function F (e.g., for monitoring convergence and for comparing solutions obtained from different initial points). Here we outline a convenient approach to computing F in practice.

The ELBO F is given by (B.19). Computing the first term is easy, and the second term is the ERSS (B.22). The third term can be computed from the marginal likelihoods L_l in (B.17), computation of which is straightforward for M_l the SER model, involving a simple sum over the p possible single effects.

Specifically we have the following lemma:

Lemma 1. *Let* $\hat{q}_l = \arg\max_q F_l(q_l, g_l, \sigma^2; \bar{\boldsymbol{r}}_l)$. *Then*

$$E_{\hat{q}_l}\left[\log\frac{g_l(\boldsymbol{\mu}_l)}{\hat{q}_l(\boldsymbol{\mu}_l)}\right] = \log L_l(\bar{\boldsymbol{r}}_l; g_l, \sigma^2) + \frac{n}{2}\log(2\pi\sigma^2) + \frac{1}{2\sigma^2}E_{\hat{q}_l}\|\bar{\boldsymbol{r}}_l - \boldsymbol{\mu}_l\|^2. \tag{B.34}$$

Proof. Rearranging (B.28) with $\boldsymbol{y}$ replaced by $\bar{\boldsymbol{r}}_l$, we have

$$\mathrm{E}_{q_l}\left[\log\frac{g_l(\boldsymbol{\mu}_l)}{q_l(\boldsymbol{\mu}_l)}\right] = F_l(q_l, g_l, \sigma^2; \bar{\boldsymbol{r}}_l) + \frac{n}{2}\log(2\pi\sigma^2) + \frac{1}{2\sigma^2}\mathrm{E}_{q_l}\|\bar{\boldsymbol{r}}_l - \boldsymbol{\mu}_l\|^2. \tag{B.35}$$

The result then follows from noting that F_l is equal to log-L_l at the optimum $q_l = \hat{q}_l$. That is, $F_l(\hat{q}_l, g_l, \sigma^2; \bar{\boldsymbol{r}}_l) = L_l(\bar{\boldsymbol{r}}_l; g_l, \sigma^2)$. □

B.6. Expression for ERSS

The expression (B.22) is derived as follows:

$$\text{ERSS} = \mathrm{E}_{\boldsymbol{q}}\left[(\boldsymbol{y} - \sum_l \boldsymbol{\mu}_l)^T(\boldsymbol{y} - \sum_l \boldsymbol{\mu}_l)\right] \tag{B.36}$$

$$= \boldsymbol{y}^T\boldsymbol{y} - 2\boldsymbol{y}^T\sum_l \bar{\boldsymbol{\mu}}_l + \sum_l\sum_{l'}\bar{\boldsymbol{\mu}}_l^T\bar{\boldsymbol{\mu}}_{l'} - \sum_l \bar{\boldsymbol{\mu}}_l^T\bar{\boldsymbol{\mu}}_l + \sum_l \mathrm{E}_{q_l}(\boldsymbol{\mu}_l^T\boldsymbol{\mu}_l) \tag{B.37}$$

$$= \|\boldsymbol{y} - \sum_l \bar{\boldsymbol{\mu}}_l\|^2 - \sum_l \bar{\boldsymbol{\mu}}_l^T\bar{\boldsymbol{\mu}}_l + \sum_l\sum_i \mathrm{E}_{q_l}[(\mu_{li})^2]. \tag{B.38}$$

Appendix C: Simulation details

C.1. Simulation data set

For the numerical simulations of eQTL fine-mapping, (Section 4), we used $n = 574$ human genotypes collected as part of the Genotype-Tissue Expression (GTEx) project (GTEx Consortium, 2017). Specifically, we obtained genotype data from whole-genome sequencing, with imputed genotypes, under dbGaP accession `phs000424.v7.p2`. In our analyses, we only included SNPs with minor allele frequencies 1% or greater. All reported SNP base-pair positions were based on Genome Reference Consortium human genome assembly 38.

To select SNPs nearby each gene, we considered two SNP selection schemes in our simulations: (i) all SNPs with 1 Megabase (Mb) of the gene's transcription start site (TSS), and (ii) the $p = 1,000$ SNPs closest to the TSS. Since the GTEx genotype data is very dense, the 1,000 closest SNPs are always less than 0.4 Mb away from the TSS, regardless of the gene considered. The first selection scheme yields genotype matrices $\boldsymbol{X}$ with at least $p = 3,022$ SNPs and at most $p = 11,999$ SNPs, with an average of $7,217$ SNPs.

C.2. CAVIAR, FINEMAP and DAP-G settings used for numerical comparisons

In CAVIAR, we set all prior inclusion probabilities to $1/p$ to match the default settings used in FINEMAP and DAP-G. In CAVIAR and FINEMAP, we set the maximum number of effect variables to the value of S that was used to simulate the gene expression data. The maximum number of iterations in FINEMAP was set to 100,000 (which is the default in FINEMAP).

All computations were performed on Linux systems with Intel Xeon E5-2680 v4 (2.40 GHz) processors. We ran *SuSiE* in R 3.5.1, with optimized matrix operations provided by the OpenBLAS dynamically linked libraries. DAP-G and CAVIAR were compiled from source using GCC 4.9.2, and pre-compiled binary executables, available from the website, were used to run FINEMAP. The result was averaged over 300 data-sets.

References

Arnold, T. and Tibshirani, R. (2016, 3). Efficient implementations of the generalized lasso dual path algorithm. *Journal of Computational and Graphical Statistics 25*(1), 1–27.

Barber, R. F. and Candès, E. J. (2015, 10). Controlling the false discovery rate via knockoffs. *Annals of Statistics 43*(5), 2055–2085.

Benner, C., Spencer, C. C., Havulinna, A. S., et al. (2016, 5). FINEMAP: efficient variable selection using summary data from genome-wide association studies. *Bioinformatics 32*(10), 1493–1501.

Bertsimas, D., King, A., Mazumder, R., et al. (2016). Best subset selection via a modern optimization lens. *Annals of Statistics 44*(2), 813–852.

Blei, D. M., Kucukelbir, A., and McAuliffe, J. D. (2017, 4). Variational inference: A review for statisticians. *Journal of the American Statistical Association 112*(518), 859–877.

Bottolo, L., Petretto, E., Blankenberg, S., et al. (2011, 12). Bayesian detection of expression quantitative trait loci hot spots. *Genetics 189*(4), 1449–1459.

Bottolo, L. and Richardson, S. (2010, sep). Evolutionary stochastic search for Bayesian model exploration. *Bayesian Analysis 5*(3), 583–618.

Carbonetto, P. and 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.

Chen, W., Larrabee, B. R., Ovsyannikova, I. G., et al. (2015, 7). Fine mapping causal variants with an approximate Bayesian method using marginal test statistics. *Genetics 200*(3), 719–736.

Chipman, H., George, E. I., and McCulloch, R. E. (2001). The practical implementation of Bayesian model selection. In *Model Selection*, Volume 38 of *IMS Lecture Notes*, pp. 65–116.

Desboulets, L. D. D. (2018). A review on variable selection in regression analysis. *Econometrics 6*(4).

Erdman, C. and Emerson, J. W. (2007). bcp: an R package for performing a Bayesian analysis of change point problems. *Journal of Statistical Software 23*(3), 1–13.

Fan, J. and Lv, J. (2010, 1). A selective overview of variable selection in high dimensional feature space. *Statistica Sinica 20*(1), 101–148.

Fraser, D. A. S. (2011, 8). Is Bayes posterior just quick and dirty confidence? *Statistical Science 26*(3), 299–316.

Freund, R. M., Grigas, P., and Mazumder, R. (2017). A new perspective on boosting in linear regression via subgradient optimization and relatives. *Annals of Statistics 45*(6), 2328–2364.

Friedman, J., Hastie, T., and Tibshirani, R. (2000). Additive logistic regression: a statistical view of boosting. *Annals of Statistics 28*(2), 337–407.

Friedman, J. H. and Stuetzle, W. (1981). Projection pursuit regression. *Journal of the American Statistical Association 76*(376), 817–823.

George, E. I. and McCulloch, R. E. (1997). Approaches for Bayesian variable selection. *Statistica Sinica 7*, 339–373.

GTEx Consortium (2017, 10). Genetic effects on gene expression across human tissues. *Nature 550*(7675), 204–213.

Guan, Y. and Stephens, M. (2011, 9). Bayesian variable selection regression for genome-wide association studies and other large-scale problems. *Annals of Applied Statistics 5*(3), 1780–1815.

Hastie, T., Tibshirani, R., and Friedman, J. (2009). *The Elements of Statistical Learning* (2 ed.). New York, NY: Springer.

Hazimeh, H. and Mazumder, R. (2018). Fast best subset selection: Coordinate descent and local combinatorial optimization algorithms. *arXiv 1803.01454*.

Hormozdiari, F., Kostem, E., Kang, E. Y., Pasaniuc, B., and Eskin, E. (2014,

10). Identifying causal variants at loci with multiple signals of association. *Genetics 198*(2), 497–508.

Huang, H., Fang, M., Jostins, L., et al. (2017, 6). Fine-mapping inflammatory bowel disease loci to single-variant resolution. *Nature 547*(7662), 173–178.

Huang, J., Breheny, P., and Ma, S. (2012, 11). A selective review of group selection in high-dimensional models. *Statistical Science 27*(4), 481–499.

Killick, R. and Eckley, I. (2014). changepoint: An R package for changepoint analysis. *Journal of statistical software 58*(3), 1–19.

Lee, Y., Francesca, L., Pique-Regi, R., and Wen, X. (2018, 5). Bayesian multi-SNP genetic association analysis: Control of FDR and use of summary statistics. *bioRxiv doi:10.1101/316471*.

Leek, J. T. and Storey, J. D. (2007). Capturing heterogeneity in gene expression studies by surrogate variable analysis. *PLoS Genetics 3*(9), e161.

Li, Y. I., van de Geijn, B., Raj, A., et al. (2016, 4). RNA splicing is a primary link between genetic variation and disease. *Science 352*(6285), 600–604.

Logsdon, B. A., Hoffman, G. E., and Mezey, J. G. (2010). A variational Bayes algorithm for fast and accurate multiple locus genome-wide association analysis. *BMC Bioinformatics 11*(1), 58.

Maller, J. B., McVean, G., Byrnes, J., et al. (2012, 12). Bayesian refinement of association signals for 14 loci in 3 common diseases. *Nature Genetics 44*(12), 1294–1301.

Mandozzi, J. and Bühlmann, P. (2016, 1). Hierarchical testing in the high-dimensional setting with correlated variables. *Journal of the American Statistical Association 111*(513), 331–343.

Meinshausen, N. (2008, 2). Hierarchical testing of variable importance. *Biometrika 95*(2), 265–278.

Meinshausen, N. and Bühlmann, P. (2010, 7). Stability selection. *Journal of the Royal Statistical Society, Series B 72*(4), 417–473.

Meuwissen, T. H., Hayes, B. J., and Goddard, M. E. (2001, 4). Prediction of total genetic value using genome-wide dense marker maps. *Genetics 157*(4), 1819–1829.

Mitchell, T. J. and Beauchamp, J. J. (1988, 12). Bayesian variable selection in linear regression. *Journal of the American Statistical Association 83*(404), 1023–1032.

Neal, R. M. (1996). *Bayesian learning for neural networks*, Volume 118 of *Lecture Notes in Statistics*. New York, NY: Springer.

Newcombe, P. J., Conti, D. V., and Richardson, S. (2016, mar). JAM: a scalable Bayesian framework for joint analysis of marginal SNP effects. *Genetic Epidemiology 40*(3), 188–201.

O'Hara, R. B. and Sillanpää, M. J. (2009, 3). A review of Bayesian variable selection methods: what, how and which. *Bayesian Analysis 4*(1), 85–117.

Olshen, A. B., Venkatraman, E. S., Lucito, R., and Wigler, M. (2004, oct). Circular binary segmentation for the analysis of array-based DNA copy number data. *Biostatistics 5*(4), 557–572.

Ott, J. (1999). *Analysis of human genetic linkage* (3 ed.). Baltimore, MD: Johns Hopkins University Press.

Pickrell, J. K. (2014). Joint analysis of functional genomic data and genome-wide association studies of 18 human traits. *American Journal of Human Genetics 94*(4), 559–573.

Schaid, D. J., Chen, W., and Larson, N. B. (2018, 8). From genome-wide associations to candidate causal variants by statistical fine-mapping. *Nature Reviews Genetics 19*(8), 491–504.

Servin, B. and Stephens, M. (2007). Imputation-based analysis of association studies: candidate regions and quantitative traits. *PLoS Genetics 3*(7), 1296–1308.

Seshan, V. E. and Olshen, A. (2018). *DNAcopy: DNA copy number data analysis.* R package version 1.56.0.

Sillanpää, M. J. and Bhattacharjee, M. (2005, 1). Bayesian association-based fine mapping in small chromosomal segments. *Genetics 169*(1), 427–439.

Spain, S. L. and Barrett, J. C. (2015, 10). Strategies for fine-mapping complex traits. *Human Molecular Genetics 24*(R1), R111–R119.

Tibshirani, R. (1996). Regression shrinkage and selection via the Lasso. *Journal of the Royal Statistical Society, Series B 58*, 267–288.

Tibshirani, R. J. (2014, 2). Adaptive piecewise polynomial estimation via trend filtering. *Annals of Statistics 42*(1), 285–323.

Urbut, S., Wang, G., Carbonetto, P., and Stephens, M. (2018). Flexible statistical methods for estimating and testing effects in genomic studies with multiple conditions. *Nature Genetics*, forthcoming.

Veyrieras, J.-B., Kudaravalli, S., Kim, S. Y., et al. (2008, oct). High-resolution mapping of expression-QTLs yields insight into human gene regulation. *PLoS Genetics 4*(10), e1000214.

Wakefield, J. (2009, jan). Bayes factors for genome-wide association studies: comparison with P-values. *Genetic Epidemiology 33*(1), 79–86.

Wallace, C., Cutler, A. J., Pontikos, N., et al. (2015, 6). Dissection of a complex disease susceptibility region using a Bayesian stochastic search approach to fine mapping. *PLOS Genetics 11*(6), e1005272.

Wang, G., Sarkar, A., Carbonetto, P., and Stephens, M. (2018, December). Code and data accompanying manuscript wang et. al (2018).

Wen, X., Lee, Y., Luca, F., and Pique-Regi, R. (2016, 6). Efficient integrative multi-SNP association analysis via deterministic approximation of posteriors. *American Journal of Human Genetics 98*(6), 1114–1129.

Zhou, X., Carbonetto, P., and Stephens, M. (2013). Polygenic modeling with Bayesian sparse linear mixed models. *PLoS Genetics 9*(2), e1003264.

Supplementary Material for "A simple new approach to variable selection in regression, with application to genetic fine-mapping"

Gao Wang[1], Abihishek Sarkar[1], Peter Carbonetto[2], and Matthew Stephens[1,3]

[1]Department of Human Genetics, The University of Chicago, Chicago, IL, 60637, USA
[2]Research Computing Center, The University of Chicago, Chicago, IL, 60637, USA
[3]Department of Statistics, The University of Chicago, Chicago, IL, 60637, USA

Supplemental figures

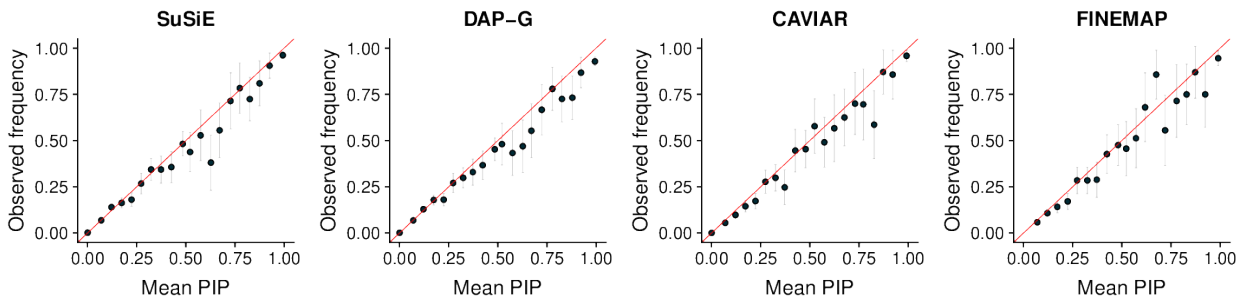


Figure S1: Assessment of PIP calibration. Variables across all simulations were grouped into bins according to their reported PIP (using 20 equal bins from 0 to 1). Shown on the plot are the mean reported PIP for each bin (X-axis) against the the empirical proportion of effect variables in that bin (Y-axis). A well calibrated method should produce points near the $y = x$ line (red). Gray vertical lines show ± 2 standard errors for the empirical proportions in each bin.

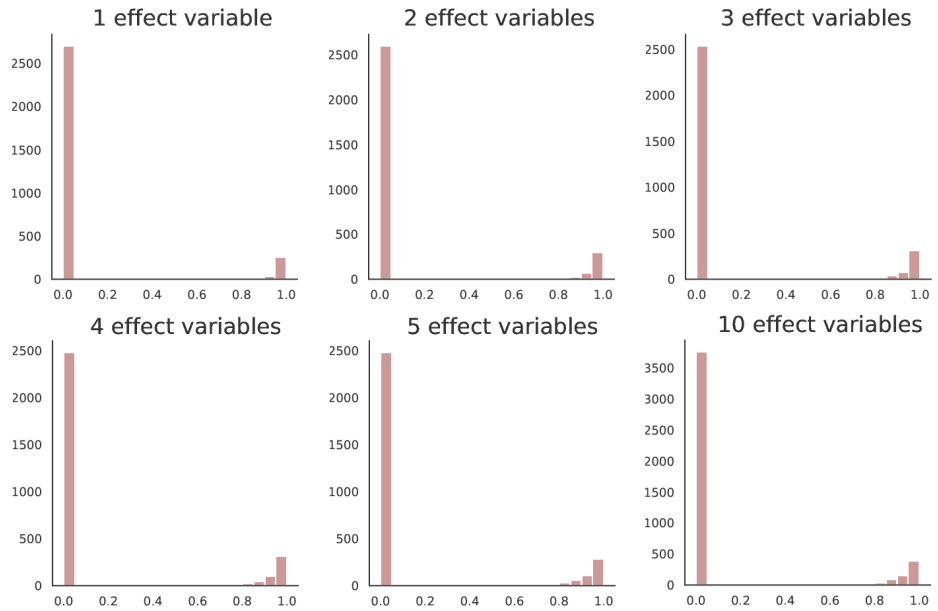


Figure S2: Distribution of purity for 95% CS sets, for different number of effect variables.

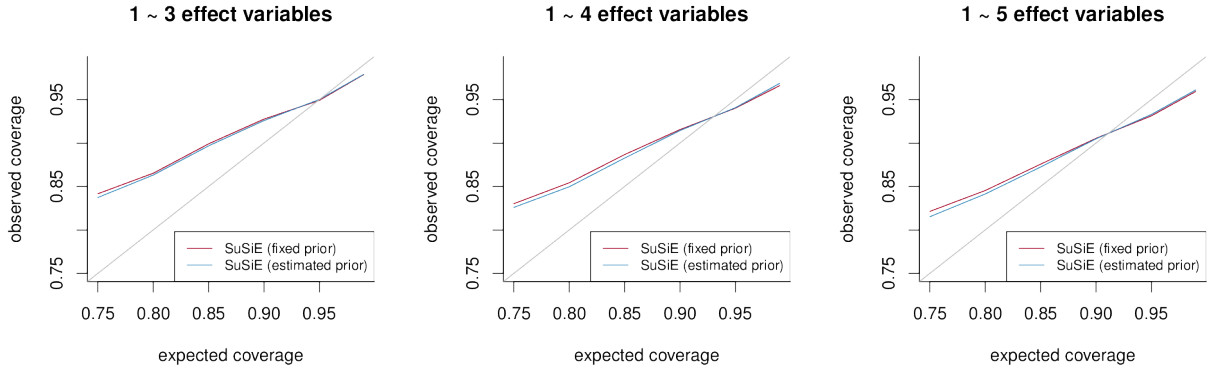


Figure S3: Assessment of CS coverage. Coverage were set to nominal levels 75%–99% (X-axis), and the corresponding empirical coverage were computed (Y-axis). Consistent with observation in Figure 3, coverage became lower as more weaker signals were analyzed.

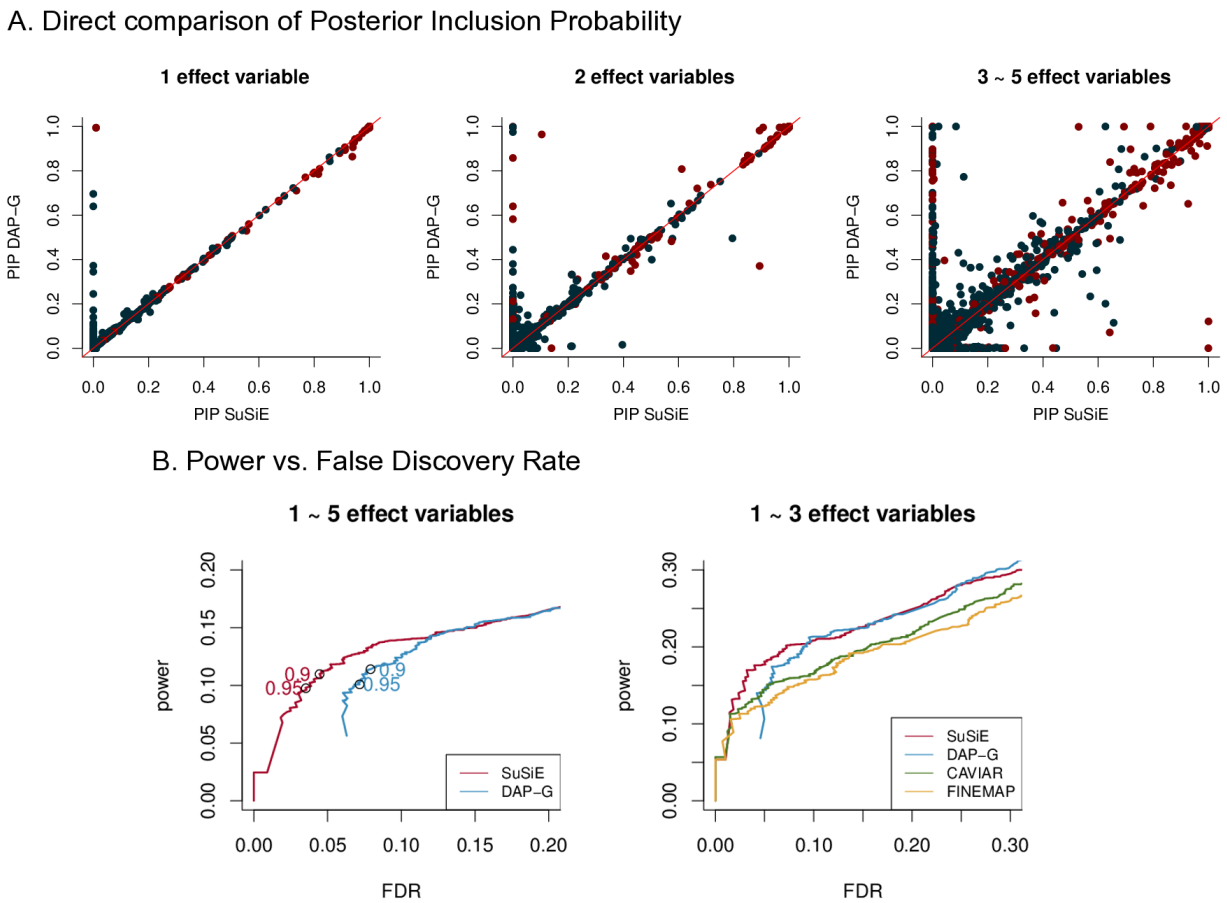


Figure S4: Comparisons of Posterior Inclusion Probabilities (PIPs) with *SuSiE* prior variance estimated. Panel A directly compares PIPs with DAP-G. **Panel B** show power vs FDR curve for different methods.

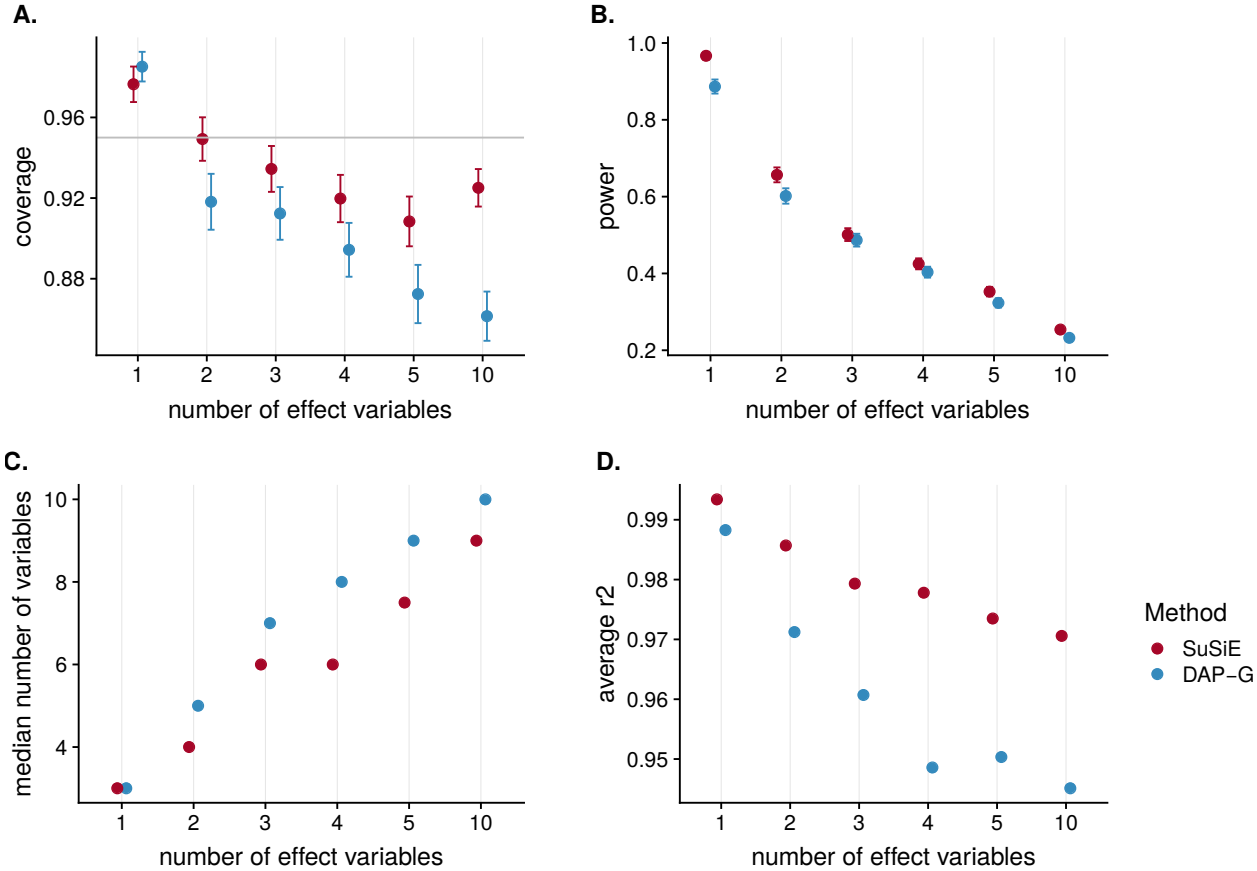


Figure S5: Comparison of 95% credible sets (CS) Same plot as Figure 3, but prior variance $\boldsymbol{\sigma_0^2}$ were estimated for *SuSiE* rather than fixing to $\sigma_{0l}^2 = 0.1$.