Bayesian averaging for variational inference applied to genomic data - First Draft

William van Rooij - EPFL

10th June 2019

Contents

Chapter 1

Introduction

1.1 Situation

For the past years, data science has been increasingly present in the world. From financial establishments to road management companies, a lot of industry sectors are integrating data science in the way business is done. With the expansion of computer performance, we are able to implement faster computation and can work with more complex models. The volume of available data, hence analysable data, is also growing, which allows more accurate inference.

Often, when trying to find a model for data, we have many more observations than parameters to fit, a *large n, small p* situation. This is the most common type of statistical analysis. Bayesian hierarchical modelling is a strong tool to identify the dependencies across multiple sources of informations, but, the number of parameters may be much larger than the number of observations. This is often the case in genomic research, where the situation is called *small n, large p*. Traditional techniques do not then apply, because of both statistical and computational constraints.

In this report, we will focus on the *small n, large p* situation in the context of genetic association. We will consider high-dimensional Bayesian inference, with its statistical advantages and its computational problem that often dissuades users to adopt this solution in statistical applications.

1.2 Motivation

Current technology allows us to numerically represent the human genome: a whole new set of data is available to study the association between the genome and various diseases or phenotypes. Some of these newly available data measure *genetic variants*, changes at specific locations on the genome (loci), the different versions of which are called *alleles*. We will focus on the most common category of genetic variants, namely, *single nucleotide polymorphism* (SNP), variations in the nucleotides that are present to some appreciable extent in the population. Some combinations of SNPs are inherited together, which yields block-wise dependence structures. We will infer associations between SNPs and transcript expression levels, called *traits*.

In Figure 1.1 are represented the correlations between real SNPs, this is the seventh chromosome from region ENm014, from Yoruba population (HapMap project) [Altshuler and Donnelly, 2005]. We can clearly see the block structure of the correlations, by the dark squares on the diagonal. Outside of the blocks, the correlations are not null but very small. A strong block correlation structure means that two SNPs in a same correlation block will be hard to differentiate. The goal is to represent the probabilities of association between a SNP and a trait, we should be able to convey the uncertainty implied by the block-wise correlation in our results.

We focus on *expression quantitative trait locus* (eQTL) analyses, which study the effects of genetic variants, in our case SNPs, on the expression of transcripts or genes. The data used for eQTL studies consist generally of several hundred thousand SNPs and thousand transcript expression outcomes. It is, in fact, a *small n, large p, large q* situation, where p is the number of SNPs, q is the number of transcripts of expressions, and n is the number of samples.

Bayesian inference involves many integrals, which usually need to be approximated. Markov Chain Monte Carlo (MCMC) algorithms are a standard technique for the approximation of integrals and can be fast and accurate when working on reasonably small datasets. When the dataset dimensions grow, however, MCMC algorithms become very time-consuming. Indeed, when performing MCMC inference, likelihoods and sometimes gradients need to be calculated at each iteration. The cost of these calculations increases with the number of parameters. Moreover, the more dimensions the problem has, the less accurate the approximations become, requiring more iterations to keep the precision needed. For the algorithm to end, all the parameters need to have converged, meaning that they all need to be checked and stored, which is often impossible when their number is very high.

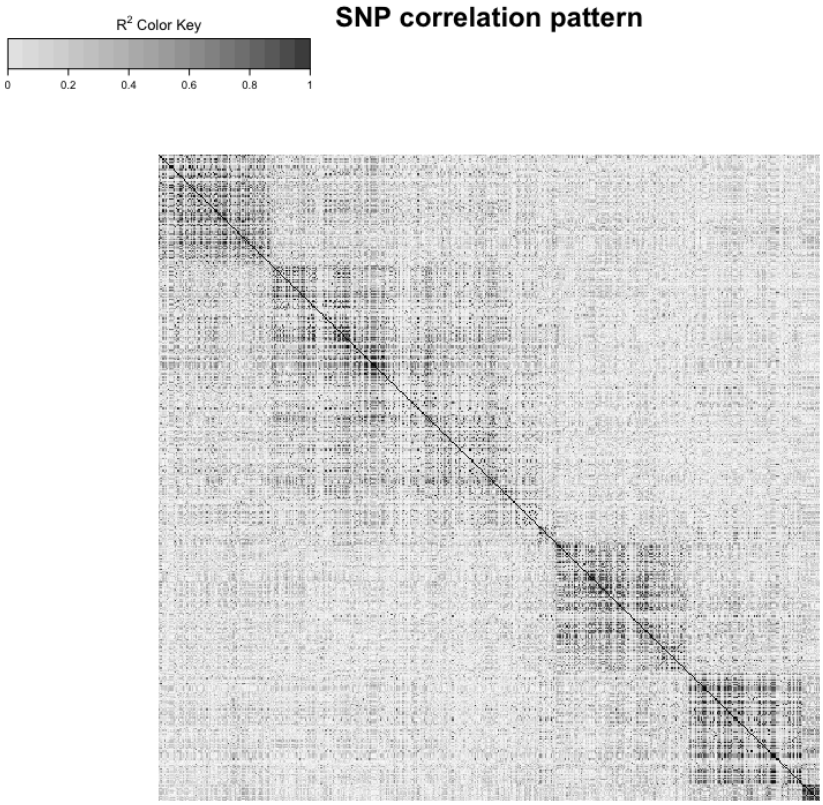

Figure 1.1: Block correlation structure of SNPs taken from Yoruba population HapMap, ENm014 region, chromosome 7 [Altshuler and Donnelly, 2005]. The darker the dot is, the stronger the correlation between the two corresponding SNPs is.

In our situation, *small n, large p, large q*, the computational cost of using an MCMC algorithm is huge. The time and memory needed to run the algorithm are not acceptable. We have to use an alternative solution, which we choose to be variational inference [David M. Blei, Alp Kucukelbir, Jon D. McAuliffe, 2018].

Chapter 2

Model

Our goal is to estimate the association between a SNP s and a trait t. To do so, we let $\boldsymbol{X} = (X_1, \ldots, X_p)$ be the design matrix, representing the SNPs, and $\boldsymbol{y} = (y_1, \ldots, y_q)$ be the response variables, representing the traits. We consider $\boldsymbol{y}$ as the response matrix and $\boldsymbol{X}$ as the candidate predictors of a hierarchical model, where each response y_t is linearly related with the predictors $\boldsymbol{X}$ and has a residual precision $\boldsymbol{\tau}_t$, i.e.,

$$\boldsymbol{y}_{n\times q} = \boldsymbol{X}_{n\times p}\, \boldsymbol{\beta}_{p\times q} + \boldsymbol{\epsilon}_{n\times q}, \quad \boldsymbol{\epsilon}_t \sim \mathcal{N}(0, \tau_t^{-1} I_n),$$

where $\boldsymbol{\beta}_{st}$ measures the association between SNP s and trait t. The parameters τ_t and σ^{-2} are assigned Gamma priors.

We introduce $\boldsymbol{\gamma}_{p\times q}$, a binary matrix to indicate which pairs of SNPs and traits are associated. The SNP s and trait t are associated if and only if $\gamma_{st} = 1$. To enforce sparsity on $\boldsymbol{\beta}$, we set a "spike-and-slab" prior distribution on β_{st}, i.e.,

$$\beta_{st} \mid \gamma_{st}, \sigma^2, \tau_t \sim \gamma_{st}\, \mathcal{N}(0, \sigma^2 \tau_t^{-1}) + (1 - \gamma_{st})\, \delta_0,$$

where δ_0 is a Dirac distribution.

The parameter ω_s controls to the proportion of responses associated with the predictor X_s, and follows a Beta distribution,

$$\omega_s \sim \text{Beta}(a_s, b_s),$$

with parameters a_s and b_s chosen to enforce sparsity. Then, the prior distribution of γ_{st} is

$$\gamma_{st} \mid \omega_s \sim \text{Bernoulli}(\omega_s).$$

If we assume $p^* \ll p$, an expected number of predictors involved in the model, we set a_s and b_s such that the prior probability that X_s is associated with at least one response is equal to p^*/p. We fix the mean of the distribution but let the variance be free, the solution still has one degree of freedom so multiple solutions are possible, e.g.,

$$a_s = 1,\ b_s = q(p - p^*)/p^*.$$

We are interested in estimating the associations between the SNPs and the traits, i.e. $\boldsymbol{\beta}$. It is common to estimate the parameters based on the observations $\boldsymbol{y}$, the associated density function is

$$\begin{aligned} p(\boldsymbol{\beta} \mid \boldsymbol{y}) &= \int \cdots \int p(\boldsymbol{\beta}, \boldsymbol{\gamma}, \boldsymbol{\omega}, \boldsymbol{\tau}, \sigma^{-2} \mid \boldsymbol{y})\ \mathrm{d}\boldsymbol{\gamma}\ \mathrm{d}\boldsymbol{\omega}\ \mathrm{d}\boldsymbol{\tau}\ \mathrm{d}\sigma^{-2}, \\ &= \frac{1}{p(\boldsymbol{y})} \int \cdots \int p(\boldsymbol{y}, \boldsymbol{\beta}, \boldsymbol{\gamma}, \boldsymbol{\omega}, \boldsymbol{\tau}, \sigma^{-2})\ \mathrm{d}\boldsymbol{\gamma}\ \mathrm{d}\boldsymbol{\omega}\ \mathrm{d}\boldsymbol{\tau}\ \mathrm{d}\sigma^{-2}, \end{aligned}$$

with

$$\begin{aligned} p(\boldsymbol{y}, \boldsymbol{\beta}, \boldsymbol{\gamma}, \boldsymbol{\omega}, \boldsymbol{\tau}, \sigma^{-2}) = &\left\{ \prod_{t=1}^{q} p(\boldsymbol{y}_t \mid \boldsymbol{\beta}_t, \tau_t) \right\} \left\{ \prod_{t=1}^{q} \prod_{s=1}^{p} p(\beta_{st} \mid \gamma_{st}, \tau_t, \sigma^{-2}) \right\} \left\{ \prod_{t=1}^{q} \prod_{s=1}^{p} p(\gamma_{st} \mid \omega_s) \right\} \\ &\times \left\{ \prod_{s=1}^{p} p(\omega_s) \right\} \left\{ \prod_{t=1}^{q} p(\tau_t) \right\} p(\sigma^{-2}), \end{aligned}$$

where, as mentioned earlier,

$$\begin{aligned} \boldsymbol{y}_t \mid \boldsymbol{\beta}_t, \tau_t &\sim \mathcal{N}_n \left(\boldsymbol{X}\boldsymbol{\beta}_t, \tau_t^{-1} \boldsymbol{I}_n \right), \\ \beta_{st} \mid \gamma_{st}, \tau_t, \sigma^{-2} &\sim \gamma_{st} \mathcal{N} \left(0, \sigma^2 \tau_t^{-1} \right) + (1 - \gamma_{st}) \delta_0, \\ \gamma_{st} \mid \omega_s &\sim \text{Bernoulli}(\omega_s), \\ \omega_s &\sim \text{Beta}(a_s, b_s), \\ \tau_t &\sim \text{Gamma}(\eta_t, \kappa_t), \\ \sigma^{-2} &\sim \text{Gamma}(\lambda, \nu), \end{aligned}$$

and δ_0 is the Dirac distribution.

Chapter 3

Variational Inference

When computing the posterior density of parameters $\boldsymbol{\theta}$ according to the observed data $\boldsymbol{y}$, variational inference simplifies the computation by approximating the posterior density $p(\boldsymbol{\theta} \mid \boldsymbol{y})$ with a simpler density $q(\boldsymbol{\theta})$. It gives an approximation to the posterior distribution as a result of an optimization problem that minimizes a measure of "closeness" as objective function.

If we have observations $\boldsymbol{y}$ and parameters $\boldsymbol{\theta}$, we need to determine the posterior distribution of the parameters conditional on the observations $p(\boldsymbol{\theta} \mid \boldsymbol{y})$. Given a family of densities $\mathcal{D}$ over the parameters, we want to find the distribution $q \in \mathcal{D}$ that minimizes the "closeness" measure compared to $p(\boldsymbol{\theta} \mid \boldsymbol{y})$.

Variational inference minimizes the Kullback–Leibler divergence as a "closeness" measure. Introduced in 1951 by Kullback and Leibler[S. Kullback and R. A. Leibler, 1951] this is the most common divergence measure used in statistics and machine learning:

$$\mathrm{KL}(q \parallel p) := \int q(\boldsymbol{\theta}) \log \left(\frac{q(\boldsymbol{\theta})}{p(\boldsymbol{\theta} \mid \boldsymbol{y})} \right) \mathrm{d}\boldsymbol{\theta}.$$

It is described as a "directed divergence" as it is asymmetric, i.e., $\mathrm{KL}(p \parallel q) \neq \mathrm{KL}(q \parallel p)$.

Determining the family $\mathcal{D}$ can be difficult, as we need the family to be simple enough to be optimized efficiently, but flexible enough for the approximation $q \in \mathcal{D}$ to be close to $p(\boldsymbol{\theta} \mid \boldsymbol{y})$ with respect to the Kullback–Leibler divergence. The approximation will then be

$$q^*(\boldsymbol{\theta}) = \underset{q(\boldsymbol{\theta}) \in \mathcal{D}}{\arg\min} \, \mathrm{KL}\left[q(\boldsymbol{\theta}) \parallel p(\boldsymbol{\theta} \mid \boldsymbol{y}) \right].$$

Minimizing the Kullback–Leibler divergence can be complicated depending on the density p that we want to approximate and the density family $\mathcal{D}$ that we want q to be part of. We can decompose the Kullback–Leibler divergence as

$$\begin{aligned}\mathrm{KL}\left[q(\boldsymbol{\theta})||p(\boldsymbol{\theta} \mid \boldsymbol{y})\right] &= \mathbb{E}\left[\log q(\boldsymbol{\theta})\right] - \mathbb{E}\left[\log p(\boldsymbol{\theta} \mid \boldsymbol{y})\right] \\ &= \mathbb{E}\left[\log q(\boldsymbol{\theta})\right] - \mathbb{E}\left[\log p(\boldsymbol{y}, \boldsymbol{\theta})\right] + \log p(\boldsymbol{y}).\end{aligned}$$

We introduce the evidence lower bound:

$$\mathcal{L}(q) = \mathbb{E}\left[\log p(\boldsymbol{\theta}, \boldsymbol{y})\right] - \mathbb{E}\left[\log q(\boldsymbol{\theta})\right] = \int q(\boldsymbol{\theta}) \log \frac{p(\boldsymbol{y}, \boldsymbol{\theta})}{q(\boldsymbol{\theta})} \mathrm{d}\boldsymbol{\theta}.$$

When decomposing the Kullback–Leibler divergence, we obtain

$$\mathrm{KL}(q \parallel p) = \log(p) - \mathcal{L}(q).$$

This means that the Kullback–Leibler divergence is the difference between the marginal log-likelihood with no effect on the optimization and a function $\mathcal{L}(q)$. Hence, minimizing the Kullback–Leibler divergence is the same as maximizing $\mathcal{L}(q)$. The difference lies in the complexity of the problems, minimizing the Kullback–Leibler divergence is not tractable, but maximizing $\mathcal{L}(q)$ admits a closed form when the family of densities $\mathcal{D}$ is well chosen. In such a case, we prefer to use $\mathcal{L}(q)$ as an objective function.

Jensen's inequality provides another way to see that $\mathcal{L}(q)$ is a lower bound for the marginal log-likelihood, which is why we call it the evidence lower bound, or variational lower bound,

$$\begin{aligned}\log p(\boldsymbol{y}) &= \log \int p(\boldsymbol{y}, \boldsymbol{\theta}) \mathrm{d}\boldsymbol{\theta}, \\ &= \log \int \frac{p(\boldsymbol{y}, \boldsymbol{\theta})}{q(\boldsymbol{\theta})} q(\boldsymbol{\theta}) \mathrm{d}\boldsymbol{\theta}, \\ &\geq \int q(\boldsymbol{\theta}) \log \left\{ \frac{p(\boldsymbol{y}, \boldsymbol{\theta})}{q(\boldsymbol{\theta})} \right\} \mathrm{d}\boldsymbol{\theta}, \\ &= \mathcal{L}(q).\end{aligned}$$

Hence, $\log p(\boldsymbol{y}) \geq \mathcal{L}(q)$.

3.1 Mean-field approximation

The complexity of the optimization problem is directly bound to the complexity of the family of densities $\mathcal{D}$ to which $q(\boldsymbol{\theta})$ belongs. We introduce the

mean-field variational family, where the parameters are mutually independent.

Let $\{\theta_j\}_{j=1}^{J}$ be a partition of $\boldsymbol{\theta}$, $q \in \mathcal{D}$ and $\mathcal{D}$ a mean-field variational family, then,

$$q(\boldsymbol{\theta}) = \prod_{j=1}^{J} q_j(\theta_j).$$

We determine the variational factors $q_j(\theta_j)$ by maximizing $\mathcal{L}(q_j)$. Hence, the variational family does not directly represent the observed data, they are both linked through the optimization of the evidence lower bound.

Concretely, we assume the independence of most of the parameters,

$$q(\boldsymbol{\theta}) = \left\{ \prod_{s=1}^{p} \prod_{t=1}^{q} q(\beta_{st}, \gamma_{st}) \right\} \left\{ \prod_{s=1}^{p} q(\omega_s) \right\} \left\{ \prod_{t=1}^{q} q(\tau_t) \right\} q(\sigma^{-2}).$$

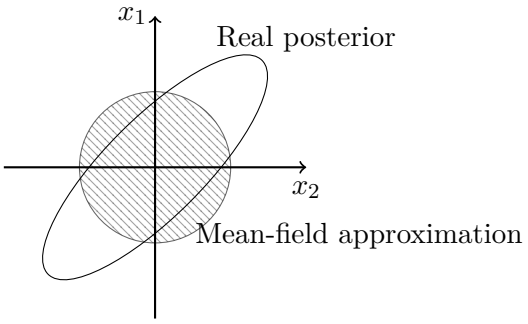

Figure 3.1: To visualise the mean-field approximation, we consider a two dimensional Gaussian distribution, represented in clear in Figure 3.1. The mean-field approximation of the posterior distribution is represented by the barred circle. We see that the mean of the approximation is the same as the real mean, but the covariance does not match the covariance of the real posterior.

We have transformed, using the evidence lower bound and the mean-field approximation our problem into a optimization problem. We now need a way to solve this problem. In the following section, we describe the coordinate ascent algorithm.

3.2 Coordinate ascent algorithm

Coordinate ascent mean-field variational inference is one of commonly used to solve this optimization problem. The algorithm iterates on the parameters

of the mean-field approximation, optimizing them one at the time. It yields a local optimum for the evidence lower bound. The algorithm is based on the following result:

Lemma 3.1 *If we fix $q_l(\theta_l)$, $l \neq j$, then the optimal $q_j^*(\theta_j)$ satisfies:*

$$q_j^*(\theta_j) \propto \exp\left\{\mathbb{E}_{-j}\left[\log p(\theta_j \mid \boldsymbol{\theta}_{-j}, \boldsymbol{y})\right]\right\}.$$

Where $\mathbb{E}_{-j}$ denotes the expectation with respect to all θ_l, $l \neq j$.

Based on this result, the algorithm updates one parameter θ_j at a time while the others stay fixed. The algorithm stops when $\mathcal{L}(q)$ increases by less than a pre-determined threshold ε.

Algorithm 1: Coordinate ascent variational inference

input : $p(\boldsymbol{y}, \boldsymbol{\theta})$, dataset y tolerance ε

output : $q(\boldsymbol{\theta}) = \prod_{j=1}^{J} q_j(\theta_j)$

initialize: $q_j(\theta_j)$

repeat

  for $j \in \{1, \ldots, J\}$ **do**

    set $q_j(\theta_j) \propto \exp\left\{\mathbb{E}_{-j}\left[\log p(\theta_j \mid \boldsymbol{\theta}_{-j}, \boldsymbol{y})\right]\right\}$

  $\mathcal{L}^{\mathsf{old}}(q) \leftarrow \mathcal{L}(q)$

  $\mathcal{L}(q) \leftarrow \mathbb{E}\left[\log p(\boldsymbol{\theta}, \boldsymbol{y})\right] - \mathbb{E}\left[\log q(\boldsymbol{\theta})\right]$

until $|\mathcal{L}^{\mathsf{old}}(q) - \mathcal{L}(q)| < \varepsilon$

return $q(\boldsymbol{\theta})$

At every iteration, $\mathcal{L}(q)$ is guaranteed to increase. The algorithm yields a local optimum depending on the initialization of the $q_j(\theta_j)$, $j = 1, \ldots, J$. Having different initializations could yield different optima that correspond to different models.

In our case, the posterior distributions of our model's parameters are:

$$\begin{aligned}
\beta_{st} \mid \gamma_{st} = 1, \boldsymbol{y} &\sim \mathcal{N}\left(\mu_{\beta,st}, \sigma^2_{\beta,st}\right), \\
\beta_{st} \mid \gamma_{st} = 0, \boldsymbol{y} &\sim \delta_0, \\
\gamma_{st} \mid \boldsymbol{y} &\sim \text{Bernoulli}(\gamma_{st}^{(1)}), \\
\omega_s \mid \boldsymbol{y} &\sim \text{Beta}(a_s^*, b_s^*), \\
\tau_t \mid \boldsymbol{y} &\sim \text{Gamma}(\eta_t^*, \kappa_t^*), \\
\sigma^{-2} \mid \boldsymbol{y} &\sim \text{Gamma}(\lambda^*, \nu^*).
\end{aligned}$$

Chapter 4

Multimodality

Bayesian model averaging is a strategy to account for multiple competing models in an inference problem. It consists of weighting the different models in a weighted average with the probability that the data corresponds to each model. The more the model corresponds to the observed data, the more it will stand out in the result.

Assume that the data $\boldsymbol{y}$ correspond to multiple models M_k, $k = 1, \ldots, K$, and Δ is the quantity of interest. We have the posterior distribution:

$$p(\Delta \mid \boldsymbol{y}) = \sum_{k=1}^{K} p(\Delta \mid M_k, \boldsymbol{y})\, p(M_k \mid \boldsymbol{y}). \tag{4.1}$$

This corresponds to a weighted average of the posterior distribution under each of the considered models with weights corresponding to the posterior models probabilities.

Instead of $p(\Delta \mid \boldsymbol{y})$ in Equation 4.1, we might be interested in approximating:

$$\mathbb{E}\left[\Delta \mid \boldsymbol{y}\right] = \sum_{k=1}^{K} \mathbb{E}\left[\Delta \mid M_k, \boldsymbol{y}\right]\ p(M_k \mid \boldsymbol{y}).$$

The posterior probability for model M_k is given by:

$$p(M_k \mid \boldsymbol{y}) = \frac{p(\boldsymbol{y} \mid M_k)\, p(M_k)}{\sum_{j=1}^{K} p(\boldsymbol{y} \mid M_j)\, p(M_j)}, \tag{4.2}$$

where $p(\boldsymbol{y} \mid M_k)$ is the likelihood of model M_k, and $p(M_k)$ is the prior probability of the model M_k. It can, for example, depend on the complexity

of the model, to favour the simpler models, or, if we consider the models to be equiprobable, it would be equal to $p(M_k) = 1/K$, $k = 1, \ldots, K$.

We know that the evidence lower bound and the Kullback–Leibler divergence are related and that minimizing the Kullback–Leibler divergence is equivalent to maximizing the evidence lower bound, and that they verify:

$$\mathrm{KL}(q \parallel p) = \log p(\boldsymbol{y}) - \mathcal{L}(q).$$

Since we minimized the Kullback–Leibler divergence, we can use $\mathcal{L}(q)$ as an approximation for $\log p(\boldsymbol{y} \mid M_k)$ in Equation 4.2.

Our quantity of interest is γ_{st}, i.e. we want to know if the SNP s and the trait t are associated. Using Algorithm 1, we initialise the distributions $q_j(\theta_j)$ with different starting points, and consider the optimums yielded by the algorithm.

We can consider each optimum to be a model representing the data, and we can apply a form of Bayesian model averaging to combine them all using the method we described here above. We approximate $\log p(\boldsymbol{y})$ by $\mathcal{L}(q)$ in Equation 4.2, and obtain an approximation for $\mathbb{E}[\gamma_{st} \mid \boldsymbol{y}]$ considering all the models we have obtained in the algorithm.

As we are dealing with strongly correlated structures, some modes will be strongly plausible even if they are not the real mode. This incertitude should be visible when observing the resulting approximations for $\mathbb{E}[\gamma_s t \mid \boldsymbol{y}]$. Indeed, we should see the real mode standing out, but we should have some other modes visible as well, more or less visible according to their plausibility.

4.1 Simulated Annealing

To identify data dependence structures, instead of altering the model, we can change the inference strategy. When dealing with highly correlated data, our coordinate ascent algorithm often gets stuck in local modes. We use a simulated annealing procedure to augment our method and improve the modes exploration.

We start with the same strategy as earlier, i.e. minimizing the reverse Kullback–Leibler divergence,

$$\mathrm{KL}(q \parallel q) = -\int q(\boldsymbol{\theta}) \log \left\{ \frac{p(\boldsymbol{\theta} \mid \boldsymbol{y})}{q(\boldsymbol{\theta})} \right\} \mathrm{d}\boldsymbol{\theta}.$$

we end up with the lower bound as objective function,

$$\mathcal{L}(q) = \mathbb{E}_q \left[\log p(\boldsymbol{y}, \boldsymbol{\theta}) \right] - \mathbb{E}_q \left[\log q(\boldsymbol{\theta}) \right],$$

which is composed of the expected log joint distribution, which motivates the approximation to take more mass where the variables explain more the data, and the entropy, that encourages the dispersion of the approximation.

The idea of simulated annealing is to introduce a temperature T that yields a series of heated distributions,

$$p_T(\boldsymbol{y}, \boldsymbol{\theta}) \propto p(\boldsymbol{y}, \boldsymbol{\theta})^{1/T},$$

and influences the differences of the modes. The temperature starts high, smoothing the density of interest, and gets lower along the process until it reaches the original density. The high temperatures yield an easier search for the global optimum. The temperature multiplies the entropy term, allowing for more disparate approximations to be considered,

$$\mathcal{L}_T(q) = \int q_T(\boldsymbol{\theta}) \log p(\boldsymbol{y}, \boldsymbol{\theta}) \mathrm{d}\boldsymbol{\theta} - T \int q_T(\boldsymbol{\theta}) \log q_T(\boldsymbol{\theta}) \mathrm{d}\boldsymbol{\theta}, \; T \geq 1, \qquad (4.3)$$

where q_T is the heated variational distribution, it applies a penalty on the log joint distribution when the temperature $T > 1$, and relaxes the penalty as T goes down until $T = 1$, where the penalty becomes null.

With the same process we used without the annealing, we can write (4.3) with respect to θ_j as

$$\mathcal{L}_T(q) = \mathbb{E}_j \left[\mathbb{E}_{-j} \left\{ \log p(\boldsymbol{y}, \boldsymbol{\theta}) \right\} - T \log q_T(\theta_j) \right] + \text{const},$$

that can be written as

$$\mathcal{L}_T(q) = T \mathbb{E}_j \left[\log \left\{ \frac{p_{T,-j}(\boldsymbol{y}, \theta_j)}{q_T(\theta_j)} \right\} \right] + \text{const},$$

where $p_{T,-j}(\boldsymbol{y}, \theta_j) \propto \exp\left\{T^{-1}\mathbb{E}_{-j}\left[\log p(\boldsymbol{y}, \boldsymbol{\theta})\right]\right\}$, $\mathbb{E}_j$ is the expected value with respect to $q_T(\theta_j)$, $\mathbb{E}_{-j}$ is the expected value with respect to every $q_T(\theta_k)$ where $k \neq j$, and const is independent of θ_j.

$\mathcal{L}_T(q)$ is maximal when $q_T(\theta_j) = p_{T,-j}(\boldsymbol{y}, \theta_j)$, which is equivalent to when

$$\log q_T(\theta_j) = T^{-1}\mathbb{E}_{-j}\left[\log p(\boldsymbol{y}, \boldsymbol{\theta})\right] + \text{const}\,, \quad j = 1, \ldots, J.$$

We have different options for the temperature schedule including a geometric spacing,

$$T_l = (1 + \Delta)^{l-1}, \quad \Delta = T_L^{1/(L-1)} - 1,$$

an harmonic spacing,

$$T_l = 1 + \Delta(l - 1), \quad \Delta\frac{T_L - 1}{L - 1},$$

and a linear spacing,

$$T_l^{-1} = T_L^{-1} + \Delta(L - l), \quad \Delta = \frac{1 - T_L^{-1}}{L - 1},$$

where $l = 1, \ldots, L$ and T_L is the hottest temperature. T_l is the temperature used at step l and L is the number of steps necessary to lower the temperature to the initial temperature $T = 1$, where the initial algorithm is ran until convergence.

To cope with strongly correlated structures and represent the incertitude of the modes, we use simulated annealing combined with our weighted average and retrieve a combination of different models yielded from different initialisations. However, two different initialisations that gave different modes could give the same mode when using simulated annealing as the density function is "smoothed" by the temperature. The number of different modes considered in the weighted average will hence be less than when not performing the simulated annealing step before hand.

Chapter 5

Simulations

Our method is based on the `locus` R-package [Ruffieux, 2019] and calls multiple times the variational algorithm before combining all the results in an weighted average. For each call, we initialize the parameters differently, and hope to obtain different optimums. Then we used the evidence lower bounds as weights in our variant of Bayesian model averaging to combine the results of each initialisation. We will call our method "multiple locus", due to the multiple calls of the variational algorithm, and "single locus", the method consisting in calling just once the algorithm.

We first tested our method on simulated data, to be able to compare the results calculated with the truth. We have used the `echoseq` R-package [Ruffieux, 2018] to generate block wise strongly autocorrelated SNPs and traits, as well as their associations. This package generates molecular quantitative data based on given association parameters.

We have generated 300 observations of 500 SNPs, with latent variables autocorrelations between 0.95 and 0.99, by blocks of 10 SNPs. It is important to note that the correlations defined here are correlations between the underlying variables, so the correlations between the SNPs are a little bit weaker. As we want to visualise the probabilities of association, we generated just one trait. We have selected five SNPs to be associated with the trait, for better visualisation, all five SNPs are in the 50 first SNPs.

In Figure 5.1, we have plotted the probabilities of association of the 50 first SNPs, out of 500 used, with a single trait t. In the construction of the data, we have enforced the five real associated SNPs to be in the 50 first SNPs, for observations reasons, they are marked in red. The real associated SNPs are the same for both of the plots.

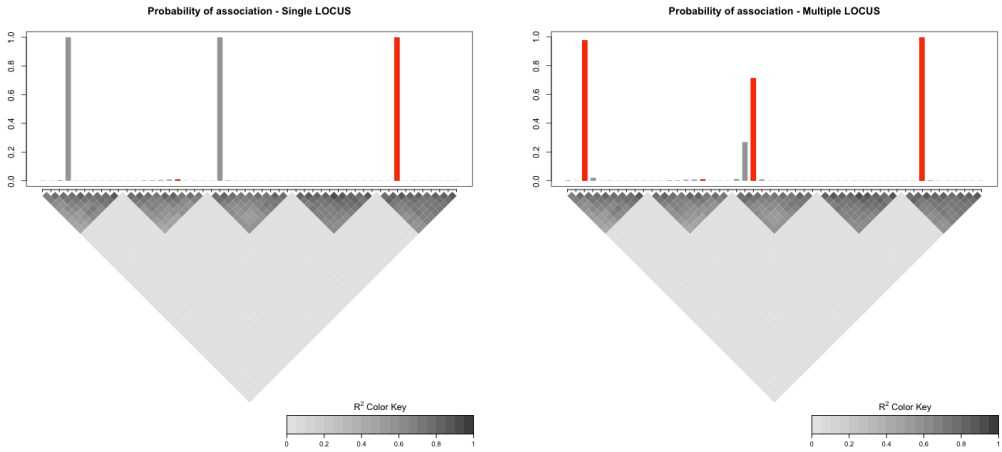

Figure 5.1: Probabilities of association of the 50 first SNPs with a single trait calculated with a single call of the `locus` function (left) and when doing a weighted average on multiple calls of the `locus` function (right). In red are the five real associated SNPs. Underneath are the correlations between the different SNPs, they are the same for the two sides as the SNPs used are the same.

On the left, we have used a single call of the `locus` function, it is equivalent to choosing a single model M and calculating

$$\mathbb{E}\left[\gamma_{st} \mid \boldsymbol{y}\right] = \mathbb{E}\left[\gamma_{st} \mid M, \boldsymbol{y}\right] \; p\left(M \mid \boldsymbol{y}\right).$$

On the right, we have used the weighted averaging method over a range of 100 different initial parameters yielding K different models M_k, $k = 1 \ldots, K$. We then calculated

$$\mathbb{E}\left[\gamma_{st} \mid \boldsymbol{y}\right] = \sum_{k=1}^{K} \mathbb{E}\left[\gamma_{st} \mid M_k\right] \; p\left(M_k \mid \boldsymbol{y}\right).$$

We can see that when using a single call of the `locus` function, the SNPs found to be associated with the trait are not the right ones. This can be explained by the strong correlations in the block structure. The strong correlations can mislead the function into yielding the wrong SNP in the same correlation block.

When using a weighted average on multiple models yielded by multiple initialisations the function `locus`, we can see that we consider different models containing the one we used to build the data. We can see that three of the five real SNPs associated with the trait are found by the algorithm.

We can also see that the two grey pikes of the left plot can be seen on the right plot. This can be explained by the fact that the model found on the left plot has been considered in the weighted average of the right plot. Apparently, a few other iterations of the function have also mislead the SNP corresponding to the second spike of the left plot to be associated with the trait, as its probability of association is non negligible. The first spike of the left plot is not as present in other occurrences of the function as we can see that it is considerably small.

The block wise correlation structure is also visible in the probabilities of association for the weighted average method. We can see that four SNPs of the middle block have all non null probabilities of association with the trait. This phenomena can be explained by the strong correlations between these SNPs misleading the algorithm into designating a wrong SNP with a strong correlation to the right one.

We compared four methods, single locus, multiple locus and their simulated annealing augmented counterparts. We chose four different situations: two of the settings involved 15 associated SNPs, whereas the remaining two had 50 associated SNPs. To ease the computation, we have chosen to consider only one trait. We also had a pair of settings where the proportion of the response variance that could be explained by the SNPs could be up to 50% and, for another pair, up to 80%. The simulated annealing augmented methods have an initial temperature fixed at $T_L = 2$, we have chosen a geometric spacing with ten steps.

In Figure 5.2 are represented the ROC curves of the four methods we wanted to compare, for each of the four settings we mentioned earlier. We have truncated the ROC curves as we are interested only in the accuracy of the methods for a small error rate. We have the same settings a we had for Figure 5.1, as well as just one trait. To fully check the accuracy of the different methods, we could calculate the association of SNPs with more traits.

Firstly, we compare the single locus and the multiple locus methods. We can clearly see in Figure 5.2 that the multiple locus is more accurate than the single locus. The multiple locus considers many different modes, in our case 100, and attributes to each mode a weight associated to the likelihood of the data being obtained from said mode. We hope that the real mode is contained in the considered 100 modes, then the likelihood of the data originating from said mode will be high and the real associated SNPs will be more represented.

Secondly, we can see that when starting both the single locus and the

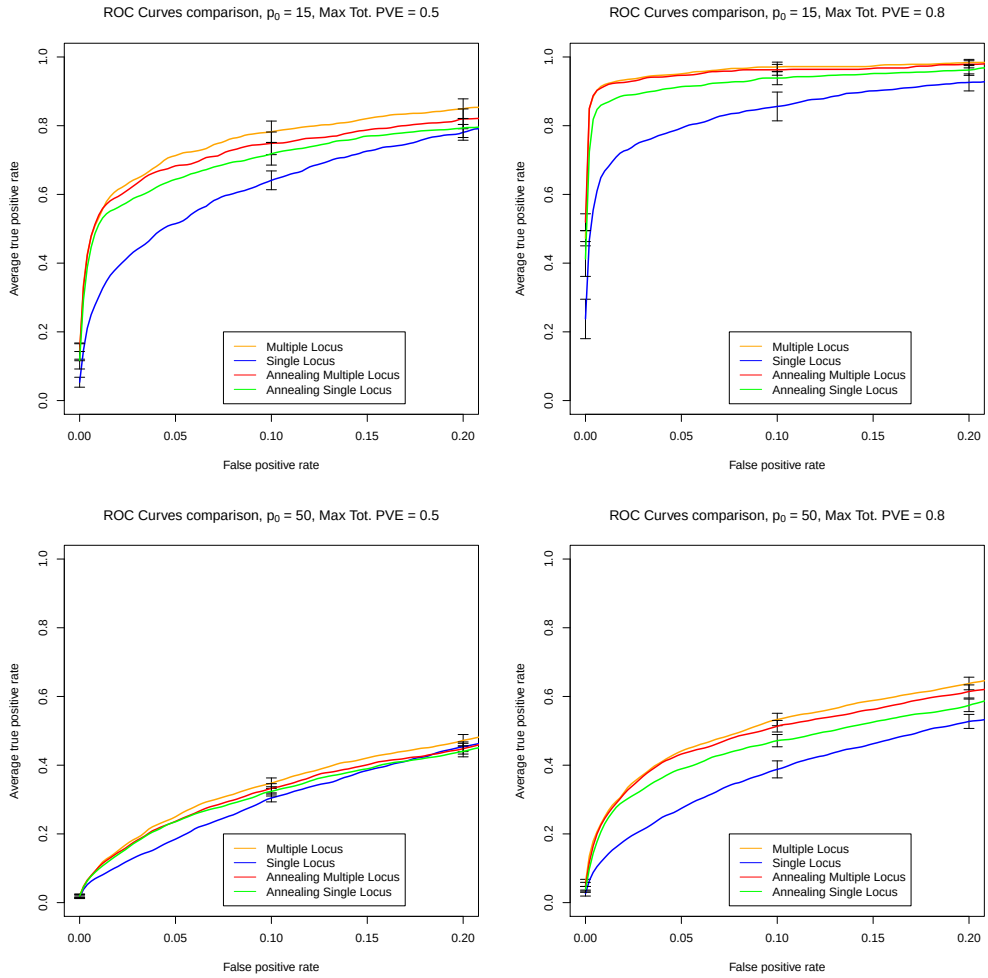

Figure 5.2: Comparison of ROC curves between multiple and single locus, and the same two methods augmented with a simulated annealing step, colored orange, blue, red, and green respectively. Top row: $p_0 = 15$, Left column: Max tot. PVE= 0.5, Bottom row: $p_0 = 50$, Right column: Max tot. PVE= 0.8

multiple locus with a simulated annealing step, the multiple locus is still more accurate than the single locus, although the difference is smaller than it is without the simulated annealing step. This means that the annealing step does not prevent the multiple locus to consider multiple different models. It also means that the simulated annealing augmented single locus does not yield the right model every time. This could be because the initial temperature we have chosen was not high enough to smooth the densities enough to access the right modes.

Thirdly, the augmented single locus method is more accurate than the single locus method. The simulated annealing step allows the method to reach modes that cannot be reached by the single locus method with certain starting parameters. This allows the real mode to be reached more often when using a simulated annealing step, hence, to be more accurate.

Fourthly, we can see that the setting where we are looking for 50 SNPs associated with the trait and a maximum proportion of the response variance explained by the SNPs is 50% have all the methods similarly accurate. The accuracy "gained" when adding the simulated annealing step or when averaging over multiple initialisations is not relevant.

Finally, we see that the augmented multiple locus is very close in accuracy to multiple locus. The advantage of starting with a simulated annealing step is not necessarily yielding more accurate results. Based on the graphs, we can even say it might be less accurate. The simulated annealing step might diminish the number of modes considered for the average, putting more weight in the wrong models, hence leading the algorithm on the wrong mode.

It should be noted that for our method, should it be simulated annealing augmented or not, paralleled computation is possible, which can drastically diminish the time needed to compute it. Even if the method has to wait until the last iteration to converge, we would still be quicker than calculating the iterations one after the other.

Instead of comparing the accuracy of different methods, we now want to assess the accuracy of our method compared to simulated values. To do so, we have generated some data with H. Ruffieux's `echoseq` `R`-package, and extracted the matrix $\boldsymbol{\beta}$ to have the real parameters. We have simulated 300 observations of four SNPs, with a equicorrelated SNPs with a strong correlation of 0.955. A strong correlation is what can induce an error in the selection of the associated SNPs, and in verifying the accuracy of our methods, it is necessary to test in extreme situations.

We compare the posterior distributions of the parameters estimations obtained from our methods with posteriors distributions obtained from MCMC inference. The two inference methods have a different convergence and stooping criteria, so the comparison should be studied prudently. Our method is based on variational inference, which has a convergence criterion defined as a tolerance to be given. The MCMC inference does not necessarily visit the whole model space, so to counter that problem, we run 10^5 iterations and burn the first half.

To be able to perform MCMC on our data, we have chosen the number

of parameters to be small, i.e. $p = 4$, $q = 1$. We are interested in evaluating the posterior distributions of $\boldsymbol{\beta} = (\beta_1, \beta_2, \beta_3, \beta_4)$. In the construction of our data, we have chosen $\beta_2, \beta_3 = 0$ and $\beta_1, \beta_4 \neq 0$.

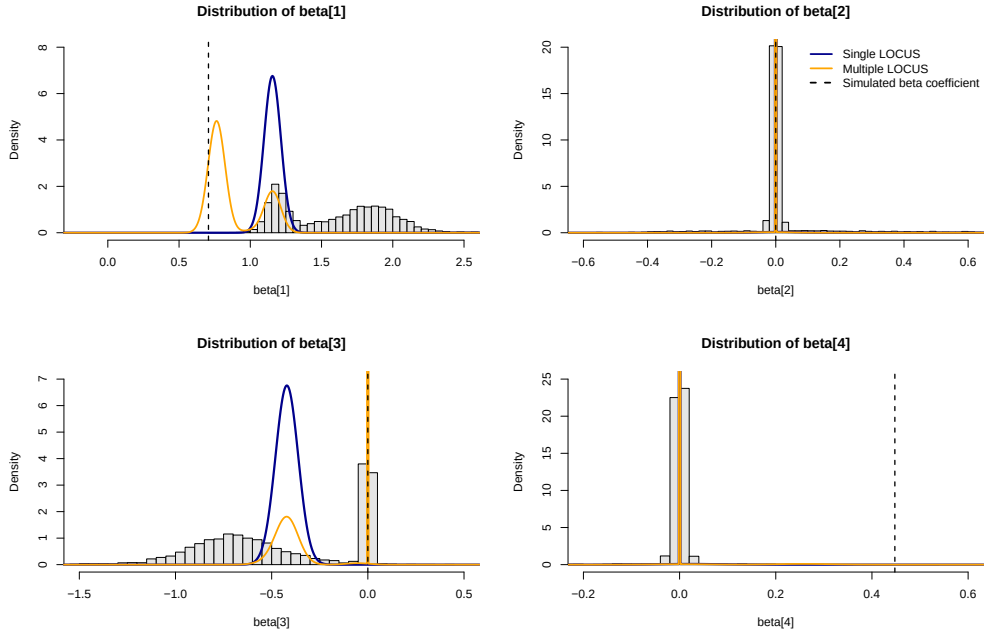

Figure 5.3: Comparison of single (blue) and multiple (orange) locus calculated posteriors for $\boldsymbol{\beta}$, MCMC simulated (histograms) $\boldsymbol{\beta}$ posteriors as well as the real (dashed black line) $\boldsymbol{\beta}$ values.

In Figure 5.3, we have plotted single and multiple locus calculated posteriors of $\boldsymbol{\beta}$, as well as the histogram of the MCMC simulated posteriors and the real values of $\boldsymbol{\beta}$. The orange and blue lines of β_2 and β_4 are superposed.

Firstly, we can see that multiple locus finds the right distribution for every β_s but once, for β_4, where it finds the same estimation as the MCMC inference and the single locus methods. However, the single locus only finds the right estimation once, for β_2. This confirms what we read in the ROC curves of Figure 5.2, where we saw that the accuracy of the multiple locus is better that the single locus.

Secondly, when the single and multiple locus do not yield the same value for the parameters, the result of single locus is visible in the distribution of multiple locus. This is given by the fact that multiple locus considers the mode obtained from single locus in its averaging, and in that case, the mode

obtained from single locus was relevant. This can be read in Figure 5.1, where we can see that the single locus selects a wrong SNP and that even if the multiple locus selects the right SNP, we can still see in the probabilities of association calculated by the multiple locus, the SNP selected by the single locus method.

Finally, β_4 is supposed to be non null, but the MCMC simulations and the approximations given by the single and multiple locus methods are all null. The real mode was probably too close to an incorrect one and the strong correlation gave the wrong mode too much weight, giving the illusion that it was the right mode. This could be caused by the "spike and slab" distribution of $\beta_{st} \mid \gamma_{st}, \sigma^2, \tau_t$, that depends on γ_{st} to be either really close to zero not. As γ_{st} must be a integer, we have to round the estimation to the closest integer. It can yield problems when the estimation of γ_{st} is close to 0.5, as a small difference in the estimation can cause a big difference in the result.

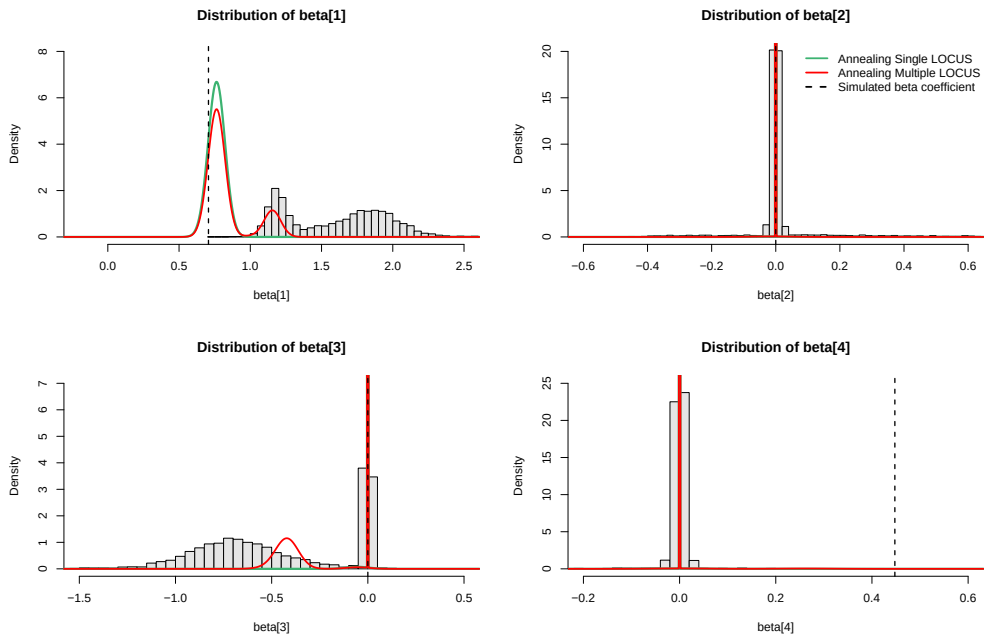

Figure 5.4: Comparison of simulated annealing augmented single (green) and multiple (red) locus calculated posteriors for $\boldsymbol{\beta}$, MCMC simulated (histograms) $\boldsymbol{\beta}$ posteriors as well as the real (dashed black line) $\boldsymbol{\beta}$ values.

In Figure 5.4 are plotted the same posteriors as in Figure 5.3, but this time we have added a simulated annealing step to the single and multiple

locus methods. We have used the same settings than for Figure 5.3, hence why the histograms and the real $\boldsymbol{\beta}$ are the same for the two situations. We have chosen an initial temperature $T_L = 5$, and we are going to perform ten geometric steps.

We can now see that for all four β_s, the single locus yields a posterior density that is more aligned with the multiple locus method. The posterior given by the single locus has strong density at the same places than the multiple locus posterior. The mode yielded by the simulated annealing augmented single locus seems to have be consequent in the weighted average of the simulated annealing augmented multiple locus.

As for the standard methods, the simulated annealing augmented methods seem to yield a correct posterior for all β_s except for β_4 where, the MCMC simulation as well as the augmented methods yield a posterior around with values condensed around zero, whereas $\beta_4 \neq 0$.

When comparing the plots of Figures 5.3 and 5.4, we can see that the annealing changed the density of the posterior yielded by the single locus method. In Figure 5.3, the density of the posterior for β_1 and β_3 were on a wrong mode, but in Figure 5.4 they match the real $\boldsymbol{\beta}$. The first simulated annealing steps helped the method get passed the local optima the algorithm yielded and reach what we believe to be the global optima of the posterior density.

Even if the simulated annealing augmented single locus posterior seem to have its values around the real values for β_1 and β_3, the simulated annealing augmented multiple locus posterior has some values where the standard single locus posterior had some values. This means that even in the simulated annealing single locus posterior we represented here has the right values, there is another initialisation of the simulated annealing augmented single locus method that yields a posterior with values around the wrong $\boldsymbol{\beta}$.

For β_1 and β_3, with the simulated annealing steps, the multiple locus method yields a posterior with a higher density on the true $\boldsymbol{\beta}$ and a lower density on the estimation yielded by the single locus method. The simulated annealing augmented single locus yielding the right $\boldsymbol{\beta}$ gives more weight in the weighted average of the multiple locus method and hence the result shows this change.

Chapter 6

Conclusion

In this paper, we wanted to build an accurate way to estimate association between SNPs and traits in a *small n, large p, large q* situation. We defined a hierarchical model linking SNPs $\boldsymbol{X} = (X_1, \ldots, X_p)$ and traits $\boldsymbol{y} = (y_1, \ldots, y_q)$, where p is the number of SNPs and q is the number of traits,

$$\boldsymbol{y} = \boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{\epsilon}, \quad \boldsymbol{\epsilon}_t \sim \mathcal{N}\left(0, \tau_t^{-1} I_n\right), \quad t = 1, \ldots, q,$$

where τ_t is the residual precision of the model, following a Gamma prior. We have, for all $t = 1, \ldots, q$, and $s = 1, \ldots, p$

$$\begin{aligned}
\boldsymbol{y}_t \mid \boldsymbol{\beta}_t, \tau_t &\sim \mathcal{N}_n\left(\boldsymbol{X}\boldsymbol{\beta}_t, \tau_t^{-1}\boldsymbol{I}_n\right), \\
\beta_{st} \mid \gamma_{st}, \tau_t, \sigma^{-2} &\sim \gamma_{st}\mathcal{N}\left(0, \sigma^2\tau_t^{-1}\right) + (1 - \gamma_{st})\delta_0, \\
\gamma_{st} \mid \omega_s &\sim \text{Bernoulli}(\omega_s), \\
\omega_s &\sim \text{Beta}(a_s, b_s), \\
\tau_t &\sim \text{Gamma}(\eta_t, \kappa_t), \\
\sigma^{-2} &\sim \text{Gamma}(\lambda, \nu),
\end{aligned}$$

where γ_{st} is an indicator of association between SNP s and trait t, and δ_0 is the Dirac distribution.

The usual tool to solve this inference problem is MCMC. However, in a *small n, large p, large q* situation, the computational cost is too high so we chose to use variational inference. It consists in approximating the posterior density $p(\boldsymbol{\theta} \mid \boldsymbol{y})$ by a simpler density $q(\boldsymbol{\theta})$ by minimizing, as a "closeness"

measure, the Kullback–Leibler divergence

$$\mathrm{KL}(q \parallel p) := \int q(\boldsymbol{\theta}) \log \left(\frac{q(\boldsymbol{\theta})}{p(\boldsymbol{\theta} \mid \boldsymbol{y})} \right) \mathrm{d}\boldsymbol{\theta}.$$

We introduced the evidence lower bound:

$$\mathcal{L}(q) = \mathbb{E}\left[\log p(\boldsymbol{\theta}, \boldsymbol{y})\right] - \mathbb{E}\left[\log q(\boldsymbol{\theta})\right] = \int q(\boldsymbol{\theta}) \log \frac{p(\boldsymbol{y}, \boldsymbol{\theta})}{q(\boldsymbol{\theta})} \mathrm{d}\boldsymbol{\theta},$$

which allows a decomposition of the Kullback–Leibler divergence:

$$\mathrm{KL}(q \parallel p) = \log(p) - \mathcal{L}(q).$$

The Kullback–leibler divergence not being tractable whereas the lower bound is, we decided to maximize the lower bound instead of minimizing the Kullback–Leibler divergence. That gives the same result thanks to the previous decomposition.

We then assumed most of the parameters independence

$$q(\boldsymbol{\theta}) = \left\{ \prod_{s=1}^{p} \prod_{t=1}^{q} q(\beta_{st}, \gamma_{st}) \right\} \left\{ \prod_{s=1}^{p} q(\omega_s) \right\} \left\{ \prod_{t=1}^{q} q(\tau_t) \right\} q(\sigma^{-2}),$$

and used the coordinate ascent mean-field algorithm to solve this optimization problem. We ended up with the following posterior distributions:

$$\begin{aligned}
\beta_{st} \mid \gamma_{st} = 1, \boldsymbol{y} &\sim \mathcal{N}\left(\mu_{\beta,st}, \sigma^2_{\beta,st}\right), \\
\beta_{st} \mid \gamma_{st} = 0, \boldsymbol{y} &\sim \delta_0, \\
\gamma_{st} \mid \boldsymbol{y} &\sim \text{Bernoulli}(\gamma^{(1)}_{st}), \\
\omega_s \mid \boldsymbol{y} &\sim \text{Beta}(a^*_s, b^*_s), \\
\tau_t \mid \boldsymbol{y} &\sim \text{Gamma}(\eta^*_t, \kappa^*_t), \\
\sigma^{-2} \mid \boldsymbol{y} &\sim \text{Gamma}(\lambda^*, \nu^*).
\end{aligned}$$

To estimate these posterior distributions, we used the `R`-package `locus`, that given the data $\boldsymbol{X}$, $\boldsymbol{y}$, and some initial parameters, uses variational inference to calculate the probabilities of association between the SNPs and the traits. We augmented this method by combining the results of multiple initialisations in a weighted average.

Our idea was that the right model exists and is reachable through a certain parameters initialisation. Soupposing we had multiple models M_k, $k = 1, \ldots, K$, we performed a weighted average on the expected value of γ_{st}, am indicator of association between SNP s and trait t

$$\mathbb{E}\left[\Delta \mid \boldsymbol{y}\right] = \sum_{k=1}^{K} \mathbb{E}\left[\Delta \mid M_k, \boldsymbol{y}\right] \; p(M_k \mid \boldsymbol{y}),$$

with

$$p(M_k \mid \boldsymbol{y}) = \frac{p(\boldsymbol{y} \mid M_k)\, p(M_k)}{\sum_{j=1}^{K} p(\boldsymbol{y} \mid M_j)\, p(M_j)},$$

where $p(\boldsymbol{y} \mid M_k)$ is the likelihood of model M_k, and $p(M_k)$ is the prior probability of model M_k.

We then used simulated annealing to try to increase the accuracy of our method. The idea is to introduce a temperature T that yields a series of heated distributions

$$p_T(\boldsymbol{y}, \boldsymbol{\theta}) \propto p(\boldsymbol{y}, \boldsymbol{\theta})^{1/T},$$

and influences the differences of the modes. The temperature starts from initial temperature T, smoothing the density of interest, and lowers along the L steps until it reaches the original density. High temperatures yield an easier search for the global optimum. The lower bound becomes

$$\mathcal{L}_T(q) = T\mathbb{E}_j \left[\log \left\{ \frac{p_{T,-j}(\boldsymbol{y}, \theta_j)}{q_T(\theta_j)} \right\} \right] + \text{const},$$

where $p_{T,-j}(\boldsymbol{y}, \theta_j) \propto \exp\left\{T^{-1}\mathbb{E}_{-j}\left[\log p(\boldsymbol{y}, \boldsymbol{\theta})\right]\right\}$, $\mathbb{E}_j$ is the expected value with respect to $q_T(\theta_j)$, $\mathbb{E}_{-j}$ is the expected value with respect to every $q_T(\theta_k)$ where $k \neq j$, and const is independent of θ_j. The lower bound is maximal when $q_T(\theta_j) = p_{T,-j}(\boldsymbol{y}, \theta_j)$, which is equivalent to when

$$\log q_T(\theta_j) = T^{-1}\mathbb{E}_{-j}\left[\log p(\boldsymbol{y}, \boldsymbol{\theta})\right] + \text{const}\,, \quad j = 1, \ldots, J,$$

where J is the number of parameters.

We compared the accuracy of four methods: the original variational implementation from the package `locus` (single locus), our weighted average augmented method (multiple locus), and their simulated annealing counterparts.

We arrived to the conclusion that the multiple locus method helps better visualise the block correlation structure when the latent variable correlations are strong, as the strong correlation induced incertitude is readable in the probabilities of association. The accuracy of the multiple locus method is better than the single locus method, but when augmenting the methods with simulated annealing, the gap gets smaller. The multiple locus method would probably reach the mode found by the single locus method, if said mode was the correct one, its weight in the average would be considerable, if the mode was not the correct one, it would have less weight than a more probable mode, hence, the better accuracy of the multiple locus method is understandable.

The simulated annealing augmented single locus method performs better than the standard single locus method, but the simulated annealing

augmented multiple locus has approximately the same accuracy than the standard multiple locus method. The reason behind this is that the standard method reaches already a lot of modes, so chances are that the right mode is reached and the weight attributed to that model will be important as the data belongs to that model.

An important point for the multiple locus method is that parallel computation is possible, so the time needed to compute the probabilities of association is greatly diminished compared to computing every iteration one after the other. The method has to wait until the last iteration as converge, but it would still be quicker.

6.1 Next steps

To be able to tell if our algorithm adequately explores the local modes, we want to represent them with the level curves similarly as V. Rockova [Rocková, 2017] did.

We plan to optimize the code that we implemented, to have an acceptable comparison with the other methods commonly used. If the results are satisfactory, we may include the function in H. Ruffieux's `R`-package (http://github.com/hruffieux/locus).

Finally, if the results are acceptable, we would like to be able to apply this method on real-life data would be the cherry on top of the cake.

Bibliography

[Altshuler and Donnelly, 2005] Altshuler, D. and Donnelly, P. (2005). A haplotype map of the human genome. *International HapMap Consortium*.

[David M. Blei, Alp Kucukelbir, Jon D. McAuliffe, 2018] David M. Blei, Alp Kucukelbir, Jon D. McAuliffe (2018). Variational inference: A review for statisticians.

[Hélène Ruffieux, 2018] Hélène Ruffieux, Anthony C. Davison, J. H. J. I. B. P. F. S. R. L. B. (2018). A global-local approach for detecting hotspots in multiple-response regression.

[Rocková, 2017] Rocková, V. (2017). Particle em for variable selection.

[Ruffieux, 2018] Ruffieux, H. (2018). *echoseq: Replication and simulation of genetic variants, molecular expression levels and other phenotypic data*. R package version 0.2.3.

[Ruffieux, 2019] Ruffieux, H. (2019). *locus: Large-scale variational inference for combined selection of covariate and response variables in regression models*. R package version 0.9.0.

[S. Kullback and R. A. Leibler, 1951] S. Kullback and R. A. Leibler (1951). On information and sufficiency. *The Annals of Mathematical Statistics*, 22:79–86.