



ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

---

# Averaged Variational Inference for Hierarchical Modelling of Genetic Association

---

*Author:*

William van Rooij

*Supervisors:*

Hélène Ruffieux

Prof. Anthony Davison

A thesis submitted for the degree of

*MSc in Applied Mathematics*

15th July 2019

## Abstract

Expression quantitative trait locus (eQTL) analyses study the effects of genetic variants on the expression of transcripts or genes. The data used generally consist of several hundred thousand genetic variants and thousands of transcript expression outcomes.

In this work, we suppose that the data follow a hierarchical regression model linking the genetic variants and the outcomes. We are then confronted with a *small  $n$ , large  $p$ , large  $q$*  situation, where  $p$  is the number of genetic variants,  $q$  is the number of expression levels, and  $n$  is the number of samples. In this situation, MCMC algorithms are not suitable for Bayesian inference as their computational cost is too large.

Here, we present a fast variational algorithm to estimate the associations between genetic variants and traits based on Ruffieux et al. [2017]. We perform a weighted average of variational estimates obtained from different parameter initialisations and augment our method with simulated annealing.

We evaluate the performance of our proposal by comparing it to existing approaches and assess its accuracy through comparisons with MCMC inference on a small problem.

The code for all our numerical experiments is freely accessible at <https://github.com/WilliamVanRooij/MasterProject>.

# Contents

<b>1</b>	<b>Introduction</b>	<b>3</b>
1.1	Situation . . . . .	3
1.2	Motivation . . . . .	3
<b>2</b>	<b>Hierarchical sparse regression for multiple responses</b>	<b>6</b>
2.1	Model . . . . .	6
2.2	Parameters of interest for variable selection . . . . .	7
<b>3</b>	<b>Variational Inference</b>	<b>8</b>
3.1	General principles . . . . .	8
3.2	Mean-field approximation . . . . .	9
3.3	Coordinate ascent . . . . .	10
<b>4</b>	<b>Multimodality</b>	<b>12</b>
4.1	Problem statement . . . . .	12
4.2	Annealed variational inference . . . . .	13
4.3	Averaged variational inference . . . . .	14
<b>5</b>	<b>Simulations</b>	<b>16</b>
5.1	Preliminary illustration . . . . .	16
5.2	Variable selection performance . . . . .	18

5.3	Comparison with MCMC inference . . . . .	25
5.4	Running times . . . . .	29
<b>6</b>	<b>Conclusion</b>	<b>31</b>

# 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 into 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, 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 valuable tool to identify the dependencies across multiple sources of information, 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 apply then, because of both statistical and computational constraints.

In this thesis, we will focus on the *small  $n$ , large  $p$*  situation in the context of genetic association. We will tackle high-dimensional regression in the Bayesian framework, with its statistical advantages and its computational problem, which often dissuades users from adopting 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 influence of the genome on 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 polymorphisms* (SNPs), i.e., 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.

Figure 1.1 shows the correlations between real SNPs, located in region ENm014 on the seventh chromosome, from a Yoruba population [Altshuler et al., 2005]. We clearly see a local block structure; outside the blocks, the correlations are not null but very small. A strong block correlation structure means that two SNPs in the same block may be statistically hard to differentiate. The goal is to represent the probabilities of association between a SNP and a trait of interest, while conveying the uncertainty implied by the block correlation in our results.

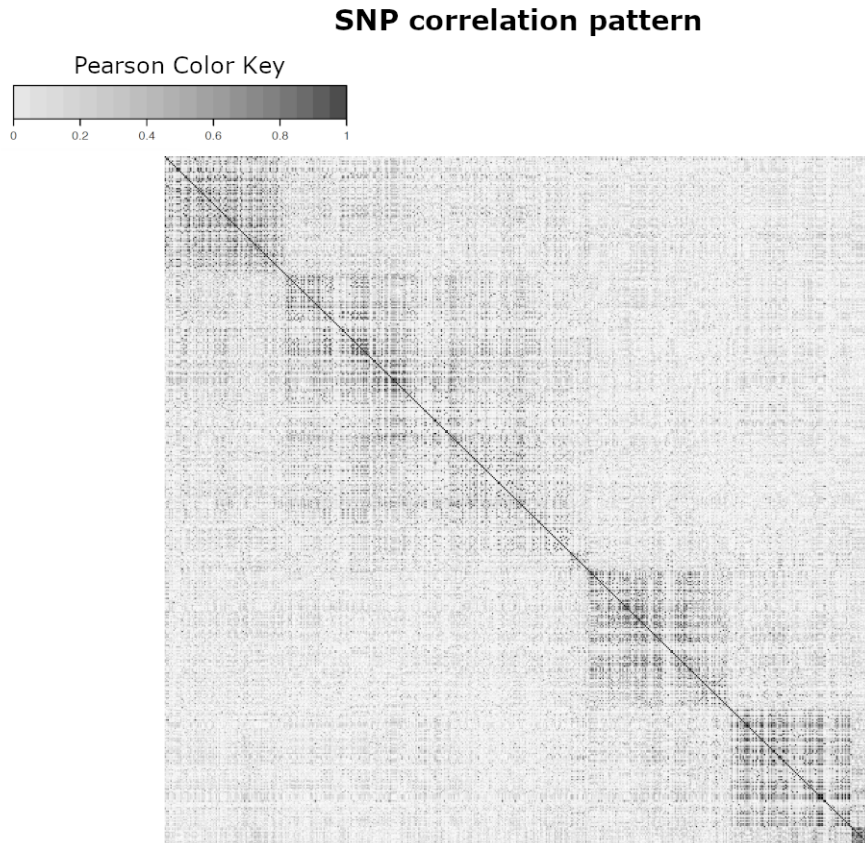


Figure 1.1: Block correlation structure of SNPs taken from a Yoruba population HapMap, ENm014 region, chromosome 7 [Altshuler et al., 2005]. The darker the dot, the stronger the correlation between the two corresponding SNPs.

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 generally consist of several hundred thousand SNPs and thousands of 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 expression outcomes, 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 tend to become very time-consuming. Indeed, when performing MCMC inference, likelihoods and sometimes gradients typically need to be calculated at each iteration, and the cost of these calculations increases with the number of parameters. Moreover, the higher the dimension, the less accurate the approximations, and more iterations are needed to reach a given precision. 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.

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 Blei et al. [2017].

## Chapter 2

# Hierarchical sparse regression for multiple responses

### 2.1 Model

Let  $\mathbf{X} = (X_1, \dots, X_p)$  be a centered design matrix, representing the candidate predictor SNPs, and  $\mathbf{y} = (y_1, \dots, y_q)$  be a centered response matrix, representing the traits. We consider a hierarchical model, where each response  $y_t$  is linearly related with the predictors  $\mathbf{X}$  and has a residual precision  $\tau_t$ , i.e.,

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

where  $\boldsymbol{\beta}$  is the matrix of regression coefficients. The parameters  $\tau_t$  and  $\sigma^{-2}$  are assigned Gamma priors.

We introduce  $\gamma_{p \times q}$ , 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  $\beta_{st}$ , 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  $\delta_0$  is the Dirac distribution.

The prior distribution of  $\gamma_{st}$  is

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

where the parameter  $\omega_s$  controls to the proportion of responses associated with the predictor  $\mathbf{X}_s$ , 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. 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  $\mathbf{X}_s$  is associated with at least one response is equal to  $p^*/p$ . As we fix the mean of the distribution but let the variance be free, there is still one degree of freedom, so multiple choices are possible, e.g.,

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

as in Castillo et al. [2015].

Parameters  $\sigma$  and  $\omega_s$  are shared across all  $q$  traits, which enables the borrowing of strength across all traits having predictors in common.

## 2.2 Parameters of interest for variable selection

We are interested in estimating the associations between the SNPs and the traits by obtaining summaries of the posterior distribution of  $\gamma$  or  $\beta$ , e.g., for the latter,

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

with

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

where, as mentioned earlier,

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

for  $t = 1, \dots, q$ ,  $s = 1, \dots, p$ , and  $\delta_0$  is the Dirac distribution.

## Chapter 3

# Variational Inference

### 3.1 General principles

Variational inference simplifies the estimation of the posterior  $p(\boldsymbol{\theta} \mid \mathbf{y})$  by approximating it with a simpler density  $q(\boldsymbol{\theta})$  in an optimisation problem that minimizes a measure of “closeness”. More precisely, given a family of densities  $\mathcal{D}$  over the parameters, we want to find the distribution  $q \in \mathcal{D}$  that is the closest to  $p(\boldsymbol{\theta} \mid \mathbf{y})$  in terms of the Kullback–Leibler divergence

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

This divergence was introduced in 1951 by Kullback and Leibler [1951] and is the most common divergence measure used in statistics and machine learning. It is described as a “directed divergence” as it is asymmetric, i.e.,

$$\text{KL}(p \parallel q) \neq \text{KL}(q \parallel p).$$

Choosing the family  $\mathcal{D}$  can be difficult, as we need it to be simple enough to enable tractable inference, but flexible enough for  $q$  to accurately represent  $p(\boldsymbol{\theta} \mid \mathbf{y})$ . The approximation will then be

$$q^*(\boldsymbol{\theta}) = \arg \min_{q(\boldsymbol{\theta}) \in \mathcal{D}} \text{KL}[q(\boldsymbol{\theta}) \parallel p(\boldsymbol{\theta} \mid \mathbf{y})].$$

As its expression involves the marginal likelihood, directly 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. For this reason, we decompose the Kullback–Leibler divergence as

$$\begin{aligned} \text{KL}[q(\boldsymbol{\theta}) \parallel p(\boldsymbol{\theta} \mid \mathbf{y})] &= \mathbb{E}[\log q(\boldsymbol{\theta})] - \mathbb{E}[\log p(\boldsymbol{\theta} \mid \mathbf{y})] \\ &= \mathbb{E}[\log q(\boldsymbol{\theta})] - \mathbb{E}[\log p(\mathbf{y}, \boldsymbol{\theta})] + \log p(\mathbf{y}), \end{aligned}$$

and introduce the “evidence lower bound” on the marginal log-likelihood:

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

i.e., we obtain

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

This means that the Kullback–Leibler divergence is the difference between the marginal log-likelihood, with no effect on the optimisation, 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 typically not tractable, but maximizing  $\mathcal{L}(q)$  admits a closed form when the family of densities  $\mathcal{D}$  is well chosen. For this reason, variational inference uses  $\mathcal{L}(q)$  as its objective function.

Jensen’s inequality provides another way to see that  $\mathcal{L}(q)$  is a lower bound for the marginal log-likelihood,

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

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

## 3.2 Mean-field approximation

The complexity of the optimisation 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 a posteriori, i.e., let  $\{\theta_j\}_{j=1}^J$  be a partition of  $\boldsymbol{\theta}$ . Then,

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

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

In our case, we assume the posterior independence of most of the parameters,

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

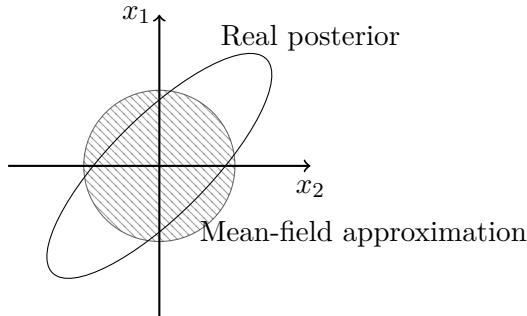


Figure 3.1: Example of the mean-field approximation, for a two-dimensional Gaussian distribution (in clear). The mean-field approximation of the posterior distribution is represented by the barred circle. The mean of the approximation agrees with the real mean, but the covariance does not match the covariance of the real posterior.

we keep  $\beta_{st}$  and  $\gamma_{st}$  grouped in order to obtain a “spike-and-slab” form a posteriori for each of the factors, rather than unimodal distributions, which would ignore the multimodal behaviour induced by the spike-and-slab prior.

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

### 3.3 Coordinate ascent

The coordinate ascent algorithm is typically used to solve the optimisation problem arising in mean-field variational inference. It iterates on the variational parameters of the mean-field approximation, optimising them one at the time and 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*

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

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

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

At every iteration,  $\mathcal{L}(q)$  is guaranteed to increase. The local optimum thus obtained may depend on the initialization of the  $q_j(\theta_j)$ ,  $j = 1, \dots, J$ ; different initializations could yield different optima that correspond to different models.

---

**Algorithm 1:** Coordinate ascent variational inference

---

**input** :  $p(\mathbf{y}, \boldsymbol{\theta})$ , dataset  $y$ , tolerance  $\varepsilon$   
**output** :  $q(\boldsymbol{\theta}) = \prod_{j=1}^J q_j(\theta_j)$   
**initialize:** the parameters of each  $q(\theta_j)$   
**repeat**  
    **for**  $j \in \{1, \dots, J\}$  **do**  
        **set**  $q_j(\theta_j) \propto \exp \{ \mathbb{E}_{-j} [\log p(\theta_j \mid \boldsymbol{\theta}_{-j}, \mathbf{y})] \}$   
         $\mathcal{L}^{\text{old}}(q) \leftarrow \mathcal{L}(q)$   
         $\mathcal{L}(q) \leftarrow \mathbb{E} [\log p(\boldsymbol{\theta}, \mathbf{y})] - \mathbb{E} [\log q(\boldsymbol{\theta})]$   
**until**  $|\mathcal{L}^{\text{old}}(q) - \mathcal{L}(q)| < \varepsilon$   
**return**  $q(\boldsymbol{\theta})$

---

For our model, the posterior distributions of our model parameters are:

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

for  $s = 1, \dots, p$ ,  $t = 1, \dots, q$ , where  $\mu_{\beta, st}$ ,  $\sigma_{\beta, st}^2$ ,  $\gamma_{st}^{(1)}$ ,  $a_s^*$ ,  $b_s^*$ ,  $\eta_t^*$ ,  $\kappa_t^*$ ,  $\lambda^*$ , and  $\nu^*$  are the “variational” parameters obtained after convergence of Algorithm 1. Their complete expression is given in Appendix B of Ruffieux [2018b].

## Chapter 4

# Multimodality

### 4.1 Problem statement

When applied to highly correlated data, variational inference underestimates posterior variances, as explained in Blei et al. [2017]. Suppose that  $p(\boldsymbol{\theta} \mid \mathbf{y})$  is the posterior distribution of  $\boldsymbol{\theta} = (\theta_1, \theta_2)$  and that we use the mean-field approximation

$$q(\boldsymbol{\theta}) = q(\theta_1)q(\theta_2).$$

As we can see in Figure 3.1, the covariance structure is altered ( $\theta_1$  and  $\theta_2$  are independent a posteriori) and the marginal variances are smaller than those of  $p(\boldsymbol{\theta} \mid \mathbf{y})$ . This also results from the optimisation of the reverse Kullback–Leibler divergence

$$\text{KL}(q \parallel p) = - \int q(\boldsymbol{\theta}) \log \frac{p(\boldsymbol{\theta} \mid \mathbf{y})}{q(\boldsymbol{\theta})} d\boldsymbol{\theta},$$

which penalizes putting mass in  $q(\cdot)$  where  $p(\cdot)$  has little mass.

The lower bound  $\mathcal{L}(q)$  tends to be highly multimodal, so the ascent algorithm (Algorithm 1) risks to get stuck in local modes. The posterior variance underestimation reinforces this risk, putting a lot of mass on one single hypothesis.

To handle this multimodality better, we will explore two routes to enhance variational inference, without changing the model. The first is to introduce a simulated annealing procedure to explore more modes; this was proposed by Ruffieux et al. [2018]. The second is to average over multiple parameter initialisations with weights equal to the posterior model probability corresponding to the obtained mode. We describe these two options in Sections 4.2 and 4.3.

## 4.2 Annealed variational inference

Simulated annealing aims at improving the exploration of multimodal parameter spaces, using heated distributions to sweep the local modes away and ease the progression to the global mode. We next describe how it can be coupled with variational inference.

We start with the same strategy as in Section 3.1, i.e., minimising the reverse Kullback–Leibler divergence,

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

and use the lower bound evidence as objective function,

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

The objective function is composed of the expected log joint distribution, which implies that the approximation will put more mass where the variables best explain the data, and the entropy, which encourages the “dispersion” of the approximation.

The idea of simulated annealing is to introduce a temperature  $T$  to obtain a series of heated distributions,

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

and control the “frequency” of the modes. The temperature starts high, smoothing the density of interest, and gets lower along the process until the original density is reached. The high temperatures facilitate the search for the global optimum. The temperature multiplies the entropy term, allowing for more disperse approximations

$$\mathcal{L}_T(q_T) = \int q_T(\boldsymbol{\theta}) \log p(\mathbf{y}, \boldsymbol{\theta}) d\boldsymbol{\theta} - T \int q_T(\boldsymbol{\theta}) \log q_T(\boldsymbol{\theta}) d\boldsymbol{\theta}, \quad T \geq 1, \quad (4.1)$$

where  $q_T$  is the heated variational distribution. Hence, annealed variational inference 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.

To obtain the annealed variational factors,  $q_T(\theta_j)$ , we write (4.1) with respect to  $\theta_j$  as

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

where  $p_{T,-j}(\mathbf{y}, \theta_j) \propto \exp \{ T^{-1} \mathbb{E}_{-j} [\log p(\mathbf{y}, \boldsymbol{\theta})] \}$ ,  $\mathbb{E}_j$  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  $\theta_j$ . The objective for  $\mathcal{L}_T(q)$  is maximal when  $q_T(\theta_j) = p_{T,-j}(\mathbf{y}, \theta_j)$ , i.e., when

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

Different choices are possible for the temperature schedule, including geometric spacing,

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

harmonic spacing,

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

and linear spacing,

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

where  $l = 1, \dots, L$  and  $T_L$  is the hottest temperature.  $T_l$  is the temperature used at step  $l$  and  $L$  is the number of steps used to lower the temperature to the initial temperature  $T = 1$ . The original variational algorithm is then run until convergence.

### 4.3 Averaged variational inference

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, accounting for the likelihood 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  $\mathbf{y}$  may have been obtained from one of multiple models  $M_k$ ,  $k = 1, \dots, K$ , and  $\Delta$  is the quantity of interest. The posterior distribution

$$p(\Delta \mid \mathbf{y}) = \sum_{k=1}^K p(\Delta \mid M_k, \mathbf{y}) p(M_k \mid \mathbf{y}) \quad (4.2)$$

corresponds to a weighted average of the posterior distribution under each of the considered models with weights corresponding to the posterior model probabilities. Instead of  $p(\Delta \mid \mathbf{y})$  in (4.2), we might be interested in summaries like the posterior mean

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

The posterior probability for model  $M_k$  is given by

$$p(M_k \mid \mathbf{y}) = \frac{p(\mathbf{y} \mid M_k) p(M_k)}{\sum_{j=1}^K p(\mathbf{y} \mid M_j) p(M_j)}, \quad (4.3)$$

where  $p(\mathbf{y} \mid M_k)$  is the likelihood under model  $M_k$ , and  $p(M_k)$  is the prior probability of model  $M_k$ . This may, for example, be chosen based on the model complexity, to favour simpler models, or, if we consider the models to be a priori equiprobable, it is set to  $p(M_k) = 1/K$ ,  $k = 1, \dots, K$ .

In Section 3.1, we saw that the evidence lower bound and the Kullback–Leibler divergence are related,

$$\text{KL}(q \parallel p) = \log p(\mathbf{y}) - \mathcal{L}(q),$$



and that minimizing the Kullback–Leibler divergence is equivalent to maximizing the evidence lower bound. Hence, by assuming that  $\mathcal{L}(q)$  is a tight lower bound on the marginal log likelihood, we can use it as an approximation for  $\log p(\mathbf{y} \mid M_k)$  in (4.3).

We propose to address the concerns described in Section 3.1 by performing a form of averaging of variational inference summaries. Namely, say that our quantity of interest is  $\gamma_{st}$ , to assess the association between SNP  $s$  and trait  $t$ . Using Algorithm 1, we initialise the distributions  $q_j(\theta_j)$  with different starting points, and consider the optimums yielded by the algorithm. If we consider that each optimum yields a model representing the data, we can apply an averaging procedure to combine them all using the method described above. We approximate  $\log p(\mathbf{y})$  by  $\mathcal{L}(q)$  in (4.3), and obtain an approximation for  $\mathbb{E}[\gamma_{st} \mid \mathbf{y}]$  considering all the models obtained through the algorithm.

To cope with the high multimodality induced by strongly correlated structures and represent the uncertainty of the modes, we use simulated annealing combined with our weighted averaging procedure and retrieve a combination of different models yielded from different initialisations. We hope that the uncertainty in the selected variables will be conveyed in the resulting approximations for  $\mathbb{E}[\gamma_{st} \mid \mathbf{y}]$ .

# Chapter 5

## Simulations

### 5.1 Preliminary illustration

In this chapter, we assess the performance of our averaged variational method on simulations. We use the `locus` R-package [Ruffieux, 2019] and call the variational algorithm multiple times before combining all the results in a weighted average. As explained in Section 4.3, we initialise the parameters differently for each call, in order to possibly obtain different optima. Then we use the evidence lower bound of the different calls as weights to combine the posterior summaries of each initialisation.

For all simulations presented in this chapter, we simulate data with very strong correlation patterns to evaluate the benefit of our method in the extreme multimodality scenarios it is designed for.

We use the `echoseq` R-package [Ruffieux, 2018a] to generate blocks of strongly auto-correlated SNPs and traits, as well as associations between them. The SNPs are coded as discrete variables describing their state and we create dependence between them using realisations of multivariate normal variables followed by a quantile thresholding rule.

For our first illustration, we generate 300 observations of 500 SNPs, by blocks of 10 SNPs, with latent variable block autocorrelations between 0.98 and 0.99. For simplicity, we simulate just one trait; the extension to multiple traits should produce similar conclusions. We select five SNPs to be associated with the trait and, for better visualisation, all five SNPs are among the 50 first SNPs.

Figure 5.1 shows the probabilities of association of the first 50 SNPs, out of 500 simulated. The LOCUS method is equivalent to choosing a single model  $M$  and calculating

$$\mathbb{E}[\gamma_{st} \mid \mathbf{y}] = \mathbb{E}[\gamma_{st} \mid M, \mathbf{y}] p(M \mid \mathbf{y}).$$

Our “averaged LOCUS” method uses a weighted average over 100 different initialisations

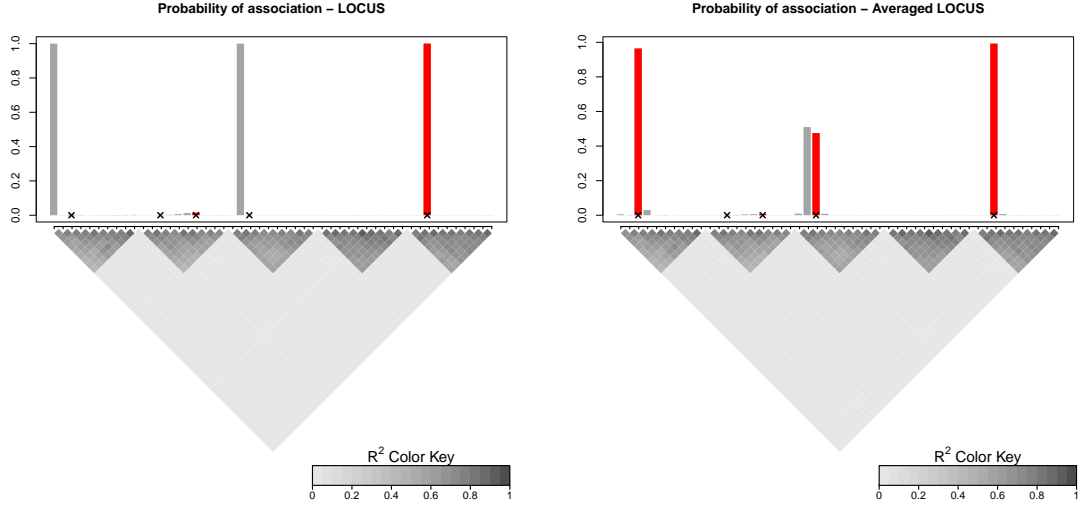


Figure 5.1: Probabilities of association of the 50 first SNPs with a single trait estimated using the original LOCUS method (left) and using our “averaged LOCUS” proposal (right), which implements the weighted averaged described in Section 4.3. In red are the five SNPs simulated as associated with the response, they are also marked with a black cross. Underneath are the extreme correlation patterns of the SNPs; they are the same for the two sides as the SNPs used are the same.

yielding 100 models  $M_k$ ,  $k = 1, \dots, 100$ :

$$\mathbb{E}[\gamma_{st} \mid \mathbf{y}] = \sum_{k=1}^{100} \mathbb{E}[\gamma_{st} \mid M_k] p(M_k \mid \mathbf{y}).$$

With the original LOCUS method, the algorithm wrongly selects two SNPs and misses four SNPs simulated as associated with the response. This can be explained by the strong correlations in the block structure creating a highly multimodal posterior and misleading the algorithm: it selected wrong SNPs among strongly correlated SNPs.

Our averaged variational inference algorithm (averaged LOCUS) does better; it identifies three of the five relevant SNPs. It also better conveys the block correlation structure in the probabilities of association as four SNPs of the middle block all have non null probabilities of association with the trait.

The second correlation block should contain two associated SNPs, and both of the methods seem to have failed to identify them as such. This could be a scaling problem. As the averaged LOCUS yields two probabilities of association almost equal to one, the smaller probabilities of association seem negligible.

Figure 5.2 shows the probabilities of association of the second block of SNPs. The probabilities of association are not all null. One of the associated SNPs has a higher probability of being associated than the others. The uncertainty resulting from the high multimodality of the problem is represented here by the probabilities of association of the

SNPs between the two associated SNPs that are all above zero. The algorithm couldn't determine which SNP was associated, the correlation between them being too strong.

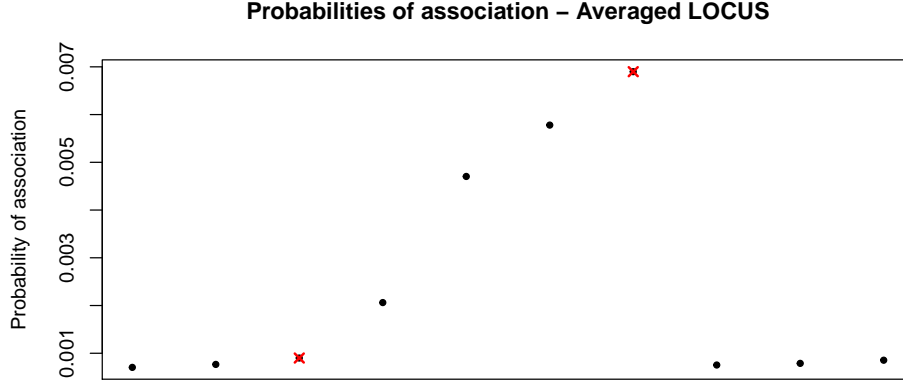


Figure 5.2: Probabilities of association for the second block of SNPs, yielded by the averaged LOCUS method. The red crosses indicate the associated SNPs.

However, as we will see in Section 5.3, when the correlations are too strong and the effect sizes are too small, our method can miss an associated SNP and compensate by adding more weight to the other SNPs. The correlation in our situation is extremely high, but if we reduce the correlation, the algorithm finds all the associated SNPs.

Figure 5.3 shows the probabilities of association of the first 50 SNPs, out of 500 simulated. The SNPs have weaker correlations as for Figure 5.1, but the same SNPs are associated with the trait. The other settings are the same as for Figure 5.1.

With weaker correlations, it is easier for LOCUS to find associated SNPs as it manages to have three actually associated SNPs, but the correlations are still too strong and mislead LOCUS into wrongfully selecting two SNPs. Averaged LOCUS, however, finds the five right associated SNPs, and we can still read the uncertainty around the middle correlation block, where a non associated SNP has a non-null probability of association.

## 5.2 Variable selection performance

In this section, we compare five methods: classical variational inference (LOCUS), averaged variational inference (averaged LOCUS), their simulated annealing augmented counterparts (annealed LOCUS and averaged annealed LOCUS), and averaged variational inference when all the weights are equal. We choose four different situations: two of the settings involve 15 associated SNPs (settings A, B), whereas the remaining two have 50

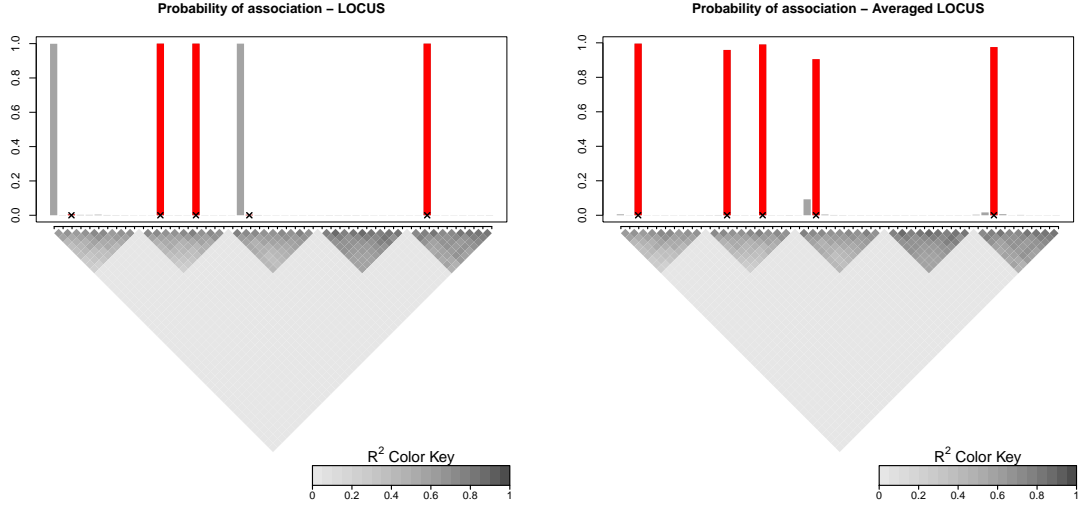


Figure 5.3: Probabilities of association of the 50 first SNPs with a single trait estimated using the original LOCUS method (left) and using our “averaged LOCUS” proposal (right), which implements the weighted averaged described in Section 4.3. In red are the five SNPs simulated as associated with the response, they are also marked with a black cross. Underneath are the correlation patterns of the SNPs, weaker than for Figure 5.1; they are the same for the two sides as the SNPs used are the same.

associated SNPs (settings C, D). For a pair of settings, the proportion of the response variance explained by the SNPs is below 50% (settings A, C) and, for another pair, below 80% (settings B, D). For simplicity, we consider only one trait. The simulated annealing augmented methods have an initial temperature fixed at  $T_L = 2$ , and a geometric spacing with ten steps. The sensitivity to these choices could be assessed in dedicated experiments. To better understand the effect of the correlations between the SNPs, Figures 5.5, 5.6, 5.7, and 5.8 have high, moderate, low, and mixed correlations respectively. We truncate the ROC curves, as we are interested only in the performance of the methods for small false positive rate.

Figure 5.5 shows the variable selection performance in terms of ROC curves for the five methods, for each of the four settings. The correlations between the SNPs are high, between 0.95 and 0.99.

First, the averaged LOCUS method clearly outperforms the LOCUS method in all four scenarios: it seems that the weighted averaging procedure effectively alleviates the risk of selecting wrong predictors in groups of highly correlated SNPs.

Second, when starting both LOCUS and averaged LOCUS with a simulated annealing step, averaged LOCUS continues to be more powerful than LOCUS, although the improvement is smaller than without simulated annealing. This suggests that the annealing step does not prevent the averaged LOCUS algorithm from selecting multiple different models, in this strongly correlated data scenario. This could be because the chosen initial temperature is not sufficiently high to smooth the densities enough to access the right

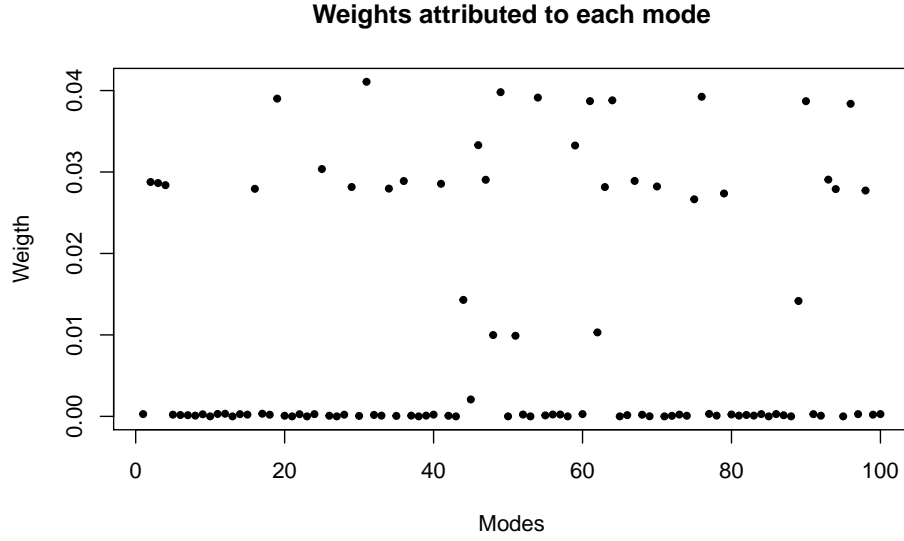


Figure 5.4: Weights attributed to the 100 obtained modes by averaged LOCUS, with correlations between SNPs between 0.95 and 0.99, 15 associated SNPs, and a proportion of the response variance explained by the SNPs below 50%.

modes.

Third, annealed LOCUS outperforms LOCUS. The simulated annealing step allows the method to reach modes that cannot be reached by the LOCUS method with certain starting parameters.

Fourth, in the less sparse setting with 50% of variance explained by the predictors (setting C), the simulated effect sizes are weaker and all methods show similar, lower, performances: the averaging or annealing procedures do not lead to much improvement.

Fifth, averaging over all the modes, without considering any weights, seems to perform better than averaged LOCUS. Figure 5.4 shows the weights attributed to each obtained mode by the averaged LOCUS method in setting A. More than half of the modes are attributed a weight equal to zero in the weighted average. Most of the modes are discarded in the averaged LOCUS method when they might have interesting information.

Finally, averaged annealed LOCUS performs similarly to averaged LOCUS: their confidence intervals overlap. In setting A, averaged annealed LOCUS might even be less powerful: the simulated annealing step might diminish the number of modes considered for the average, putting more weight on wrong models.

Figure 5.6 shows the variable selection performance in terms of the ROC curves for the five methods, for the same settings (A, B, C, D) than for Figure 5.5. The correlations between the SNPs are between 0.7 and 0.95.

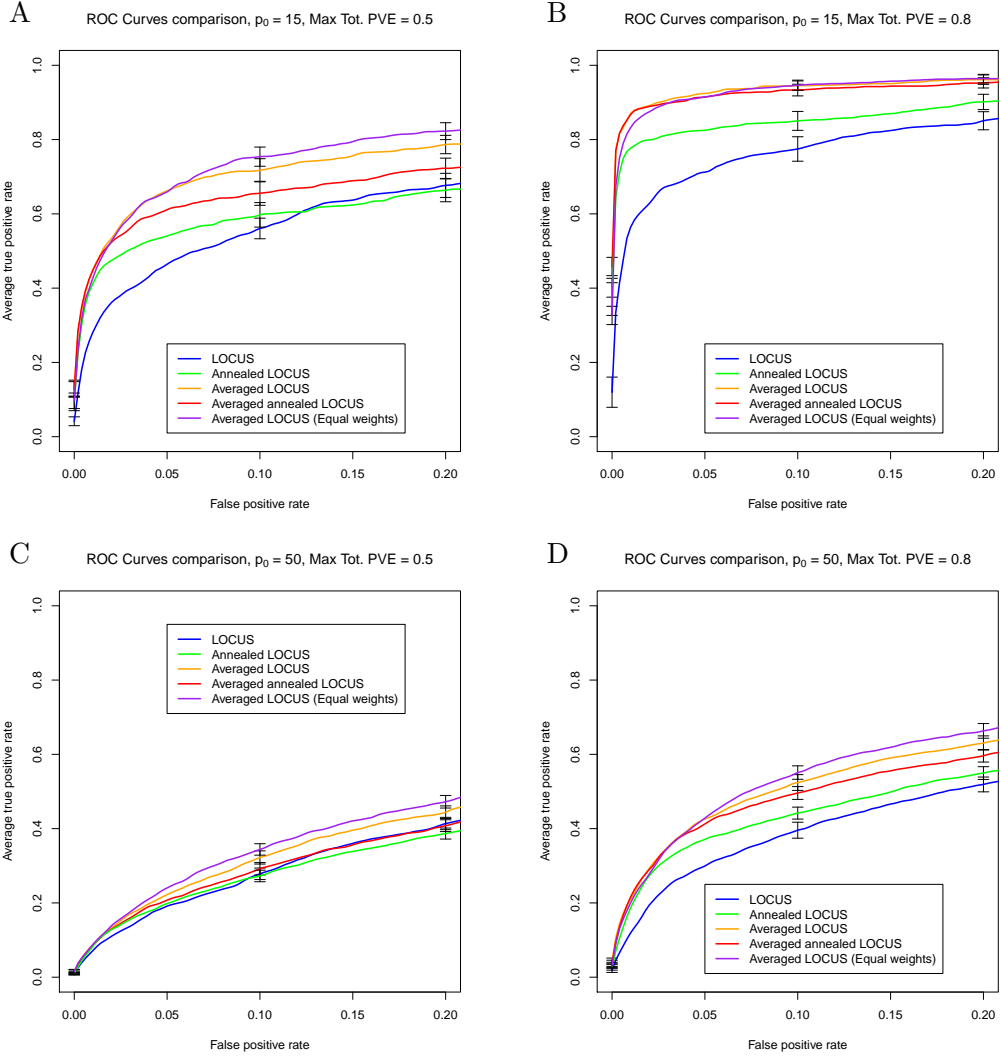


Figure 5.5: Comparison of ROC curves between LOCUS, averaged LOCUS, the same two methods augmented with a simulated annealing step, and averaged LOCUS with equal weights for every initialisation, colored orange, blue, red, green, and purple respectively. Top row:  $p_0 = 15$ , Left column: Max tot. PVE= 0.5, Bottom row:  $p_0 = 50$ , Right column: Max tot. PVE= 0.8. The correlations between the SNPs are between 0.95 and 0.99.

First, LOCUS is outperformed by the four other methods in settings A, B, and D. In setting C, the five methods seem to perform similarly. The correlations seem to be high enough to mislead LOCUS, but low enough so that averaging over multiple modes does not perform better than annealing LOCUS i.e., finding a single mode.

Second, annealed LOCUS performs better than LOCUS in all settings, the annealing helps the method find a more suitable mode. However, in setting C, their performance is similar.

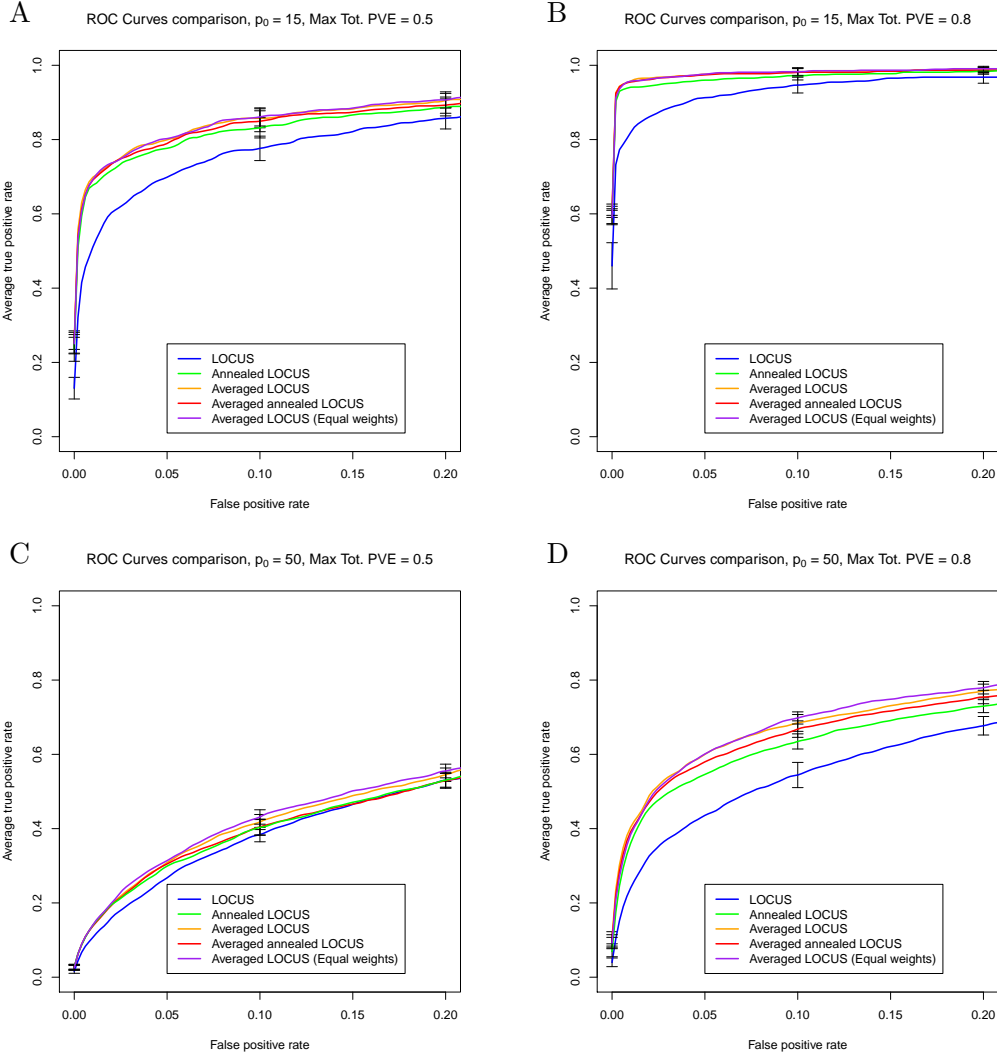


Figure 5.6: Comparison of ROC curves between LOCUS, averaged LOCUS, the same two methods augmented with a simulated annealing step, and averaged LOCUS with equal weights for every initialisation, colored orange, blue, red, green, and purple respectively. Top row:  $p_0 = 15$ , Left column: Max tot. PVE= 0.5, Bottom row:  $p_0 = 50$ , Right column: Max tot. PVE= 0.8. The correlations between the SNPs are between 0.7 and 0.95.

Finally, averaged LOCUS, with or without weights, annealed LOCUS, and averaged annealed LOCUS perform similarly in all settings. The correlation between the SNPs doesn't make the averaging over multiple models worth doing.

Figure 5.7 shows the ROC curves, representing the variable selection performance of the five methods, for the four settings (A, B, C, D), as in Figure 5.5. The correlations between the SNPs are between 0.5 and 0.7.



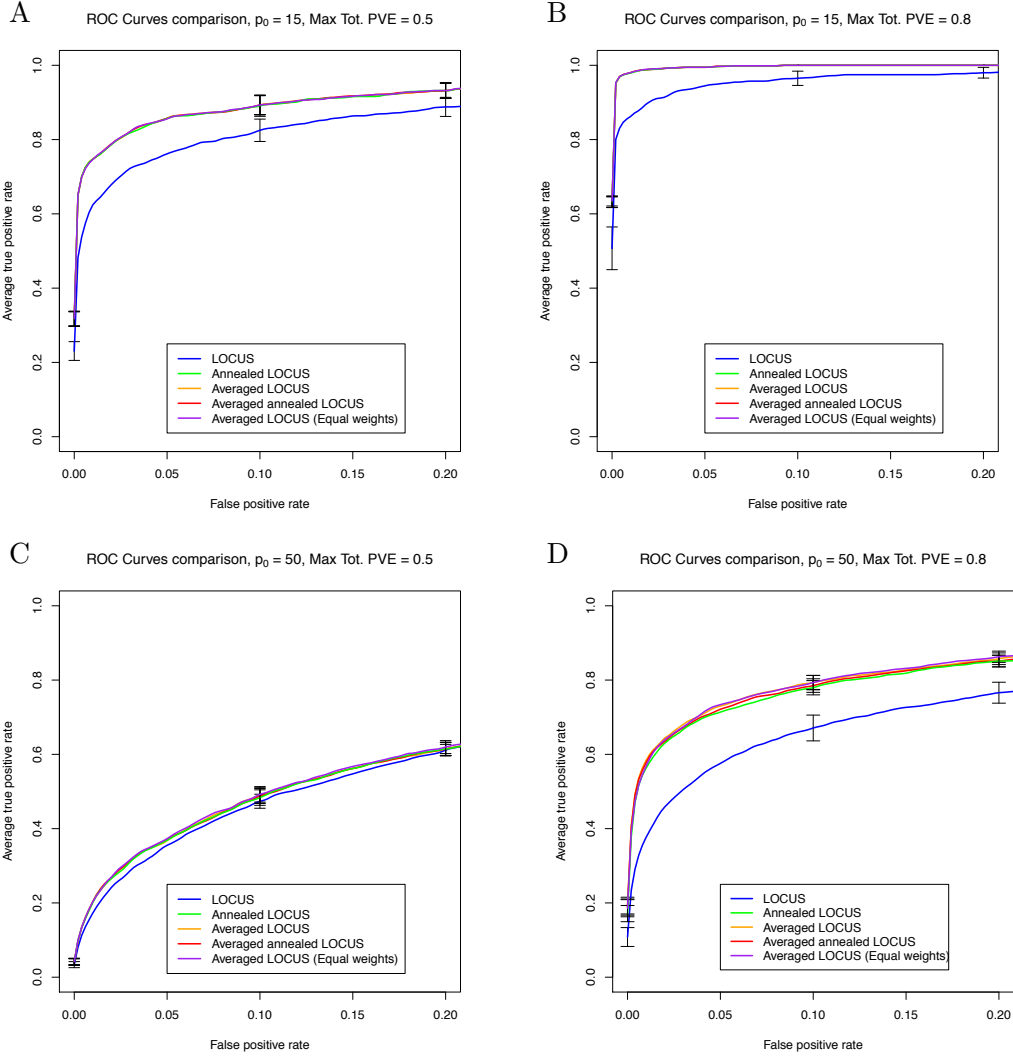


Figure 5.7: Comparison of ROC curves between LOCUS, averaged LOCUS, the same two methods augmented with a simulated annealing step, and averaged LOCUS with equal weights for every initialisation, colored orange, blue, red, green, and purple respectively. Top row:  $p_0 = 15$ , Left column: Max tot. PVE= 0.5, Bottom row:  $p_0 = 50$ , Right column: Max tot. PVE= 0.8. The correlations between the SNPs are between 0.5 and 0.7.

LOCUS is outperformed by the other methods in settings A, B, and D, as in 5.6. In setting C, all five methods seem to have a similar performance.

Then, the performances of averaged LOCUS, annealed LOCUS, averaged annealed LOCUS, and averaged LOCUS with equal weights are indistinguishable. They perform extremely similarly. The correlations between the SNPs are weak enough to able the annealing to find a mode as suited for the data as the averaging methods do.

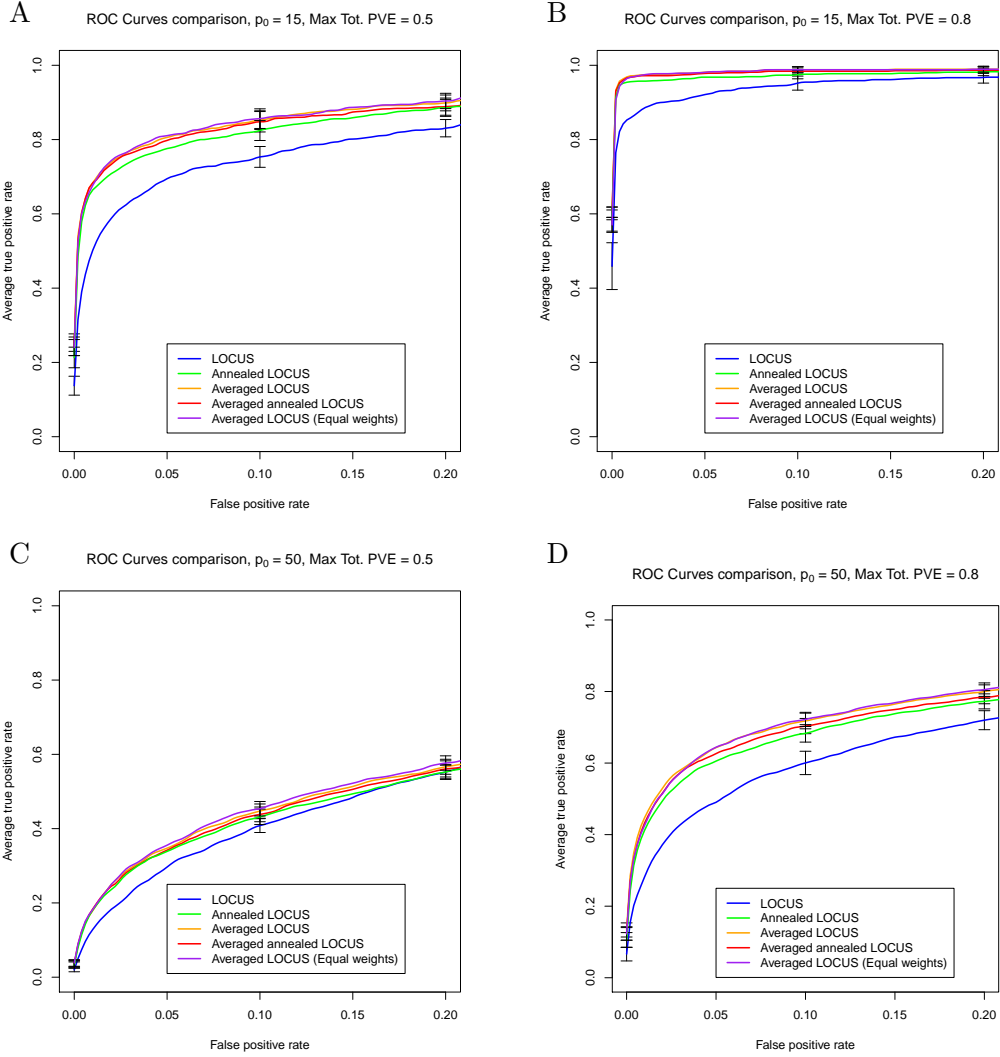


Figure 5.8: Comparison of ROC curves between LOCUS, averaged LOCUS, the same two methods augmented with a simulated annealing step, and averaged LOCUS with equal weights for every initialisation, colored orange, blue, red, green, and purple respectively. Top row:  $p_0 = 15$ , Left column: Max tot. PVE= 0.5, Bottom row:  $p_0 = 50$ , Right column: Max tot. PVE= 0.8. The correlations between the SNPs are between 0.5 and 0.99.

Figure 5.8 shows variable selection performances as ROC curves for the five methods, in four settings (A, B, C, D). The correlations between the SNPs are between 0.5 and 0.99, representing a mixture of the three previous experiments. The correlations in Figure 5.5 are extreme, we should have results closer to those of Figures 5.6 and 5.7.

LOCUS is outperformed by the four other methods in settings A, B, and D, but its performance seems to be similar to the four others in setting C. This goes with what we read in Figures 5.5, 5.6, and 5.7.

Averaged LOCUS, with or without weights, annealed LOCUS, and averaged annealed LOCUS have a similar performance. The weights do not improve the performance of the averaged LOCUS. These observations are similar to what we observed in Figures 5.6 and 5.7.

The correlations being between 0.5 and 0.99, they have a higher chance of being in the spectrum of the correlations of Figures 5.6 and 5.7 than being in the spectrum of the correlations of Figure 5.5. The behaviour of the mixed correlations performances being similar to the performances of the weak and medium correlations makes sense.

### 5.3 Comparison with MCMC inference

Section 5.2 evaluated variable selection performance of the different methods, we now compare the accuracy of our proposal by confronting it with MCMC inference. To do so, we generate data with the `echoseq` R-package, and save the simulated matrix  $\beta$ . We simulate 300 observations of four equicorrelated SNPs with an extremely high correlation coefficient of 0.995. Out of the four SNPs, two of them are associated with a trait.

We compare the posterior distributions of the regression coefficient obtained by our methods with the posterior distributions obtained by MCMC inference. The two inference methods have a different convergence and stopping 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 alleviate this problem, we run it for a large number of iterations, namely  $10^5$  iterations and discard the first half as burn-in, and we consider a very small problem, i.e.,  $p = 4, q = 1$ . We are interested in evaluating the posterior distributions of  $\beta = (\beta_1, \beta_2, \beta_3, \beta_4)$ . In the construction of our data, we have chosen  $\beta_2, \beta_3 = 0$  and  $\beta_1, \beta_4 \neq 0$ .

Figure 5.9 shows LOCUS and averaged LOCUS estimated posteriors of  $\beta$ , as well as the histograms of the MCMC posteriors and the simulated values of  $\beta$ .

First, the problem appears to be very difficult as all methods fail to accurately capture the simulated values; even the MCMC algorithm yields inferences far from the truth for  $\beta_4$ . It is supposed to be non-null, but the MCMC approximations and those given by LOCUS and averaged LOCUS are all concentrated around zero. The correlations are too strong and the effect sizes too weak for the algorithms to find the simulated  $\beta_4$ . To verify this affirmation, Figure 5.10 shows a case with weaker correlations and stronger effect sizes.

Second, averaged LOCUS probably best reflects the true posterior; it puts mass near the simulated values of  $\beta_s$  for every  $\beta_s$  but for  $\beta_4$ , where it finds the same estimation as the MCMC inference and the LOCUS methods. This is in line with the ROC curves of Figure 5.5, where we saw that for variable selection, averaged LOCUS outperforms LOCUS.

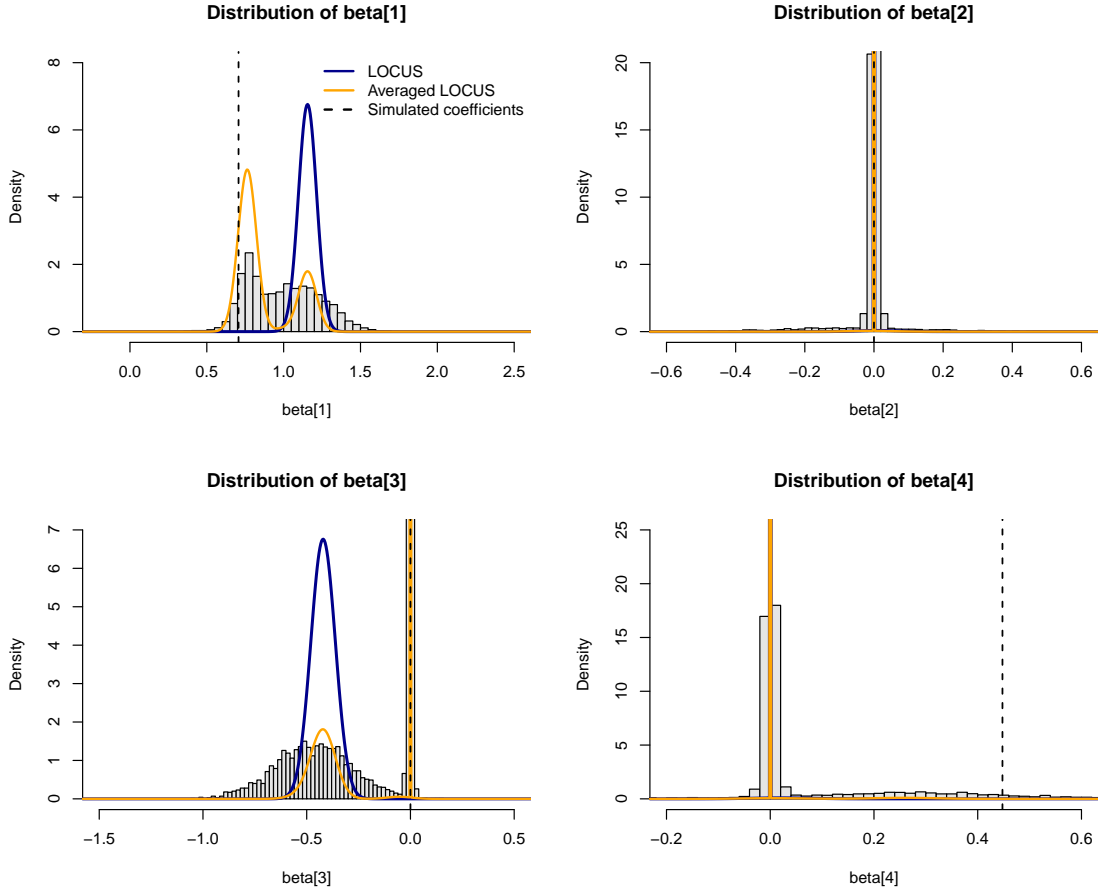


Figure 5.9: Comparison of LOCUS (blue) and averaged LOCUS (orange) estimated posterior distributions for  $\beta$ , MCMC distributions (histograms), as well as the simulated  $\beta$  values (dashed black line). The orange and blue lines of  $\beta_2$  and  $\beta_4$  are superimposed. The SNPs are equicorrelated with a correlation coefficient of 0.995.

Third, when LOCUS and averaged LOCUS disagree, the result of LOCUS is “visible” in the distribution of averaged LOCUS. Averaged LOCUS considers the mode obtained from LOCUS in its averaging.

Finally, the shift between the simulated  $\beta_1$  and the algorithms yields is in the wrong way. It is normal to have a shrinkage when estimating simulated values with inference, but the shift is supposed to be in the direction of zero. The algorithms, both MCMC, LOCUS, and averaged LOCUS, having wrongly not selected the fourth SNP as associated, they certainly compensated on  $\beta_1$ , hence why the estimated effect size  $\beta_1$  is bigger than the simulated  $\beta_4$ .

Figure 5.10 shows LOCUS and averaged LOCUS estimated posteriors of  $\beta$ , the histograms of the MCMC posteriors, and the simulated values of  $\beta$ , for SNPs autocorrelated

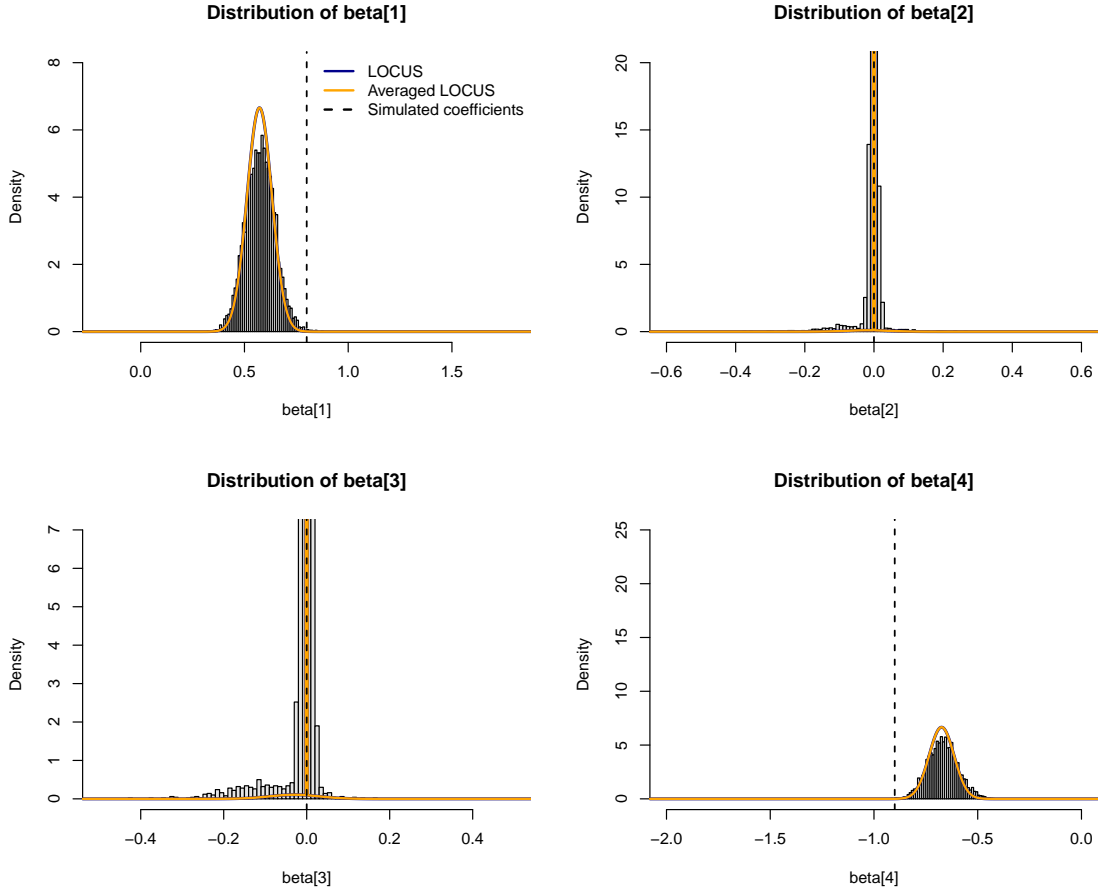


Figure 5.10: Comparison of LOCUS (blue) and averaged LOCUS (orange) estimated posterior distributions for  $\beta$ , MCMC distributions (histograms), as well as the simulated  $\beta$  values (dashed black line). The orange and blue lines are superimposed. The SNPs are autocorrelated with a correlation coefficient of 0.9. The blue and orange lines are superimposed.

with a correlation coefficient of 0.9, weaker than for Figure 5.9, and a stronger effect size for  $\beta_4$ .

The weaker correlations and stronger effect size for  $\beta_4$  allows the algorithms to find a better estimation of the simulated effect sizes  $\beta$ . The active SNPs have been correctly estimated, and the inactive SNPs have been detected.

LOCUS and averaged LOCUS yield similar posterior distributions for  $\beta$ . Due to the correlation being weaker and the effect sizes being bigger, LOCUS can estimate the effect sizes just as well as averaged LOCUS does.

Figure 5.11 shows the same posteriors as Figure 5.9, but with a simulated annealing

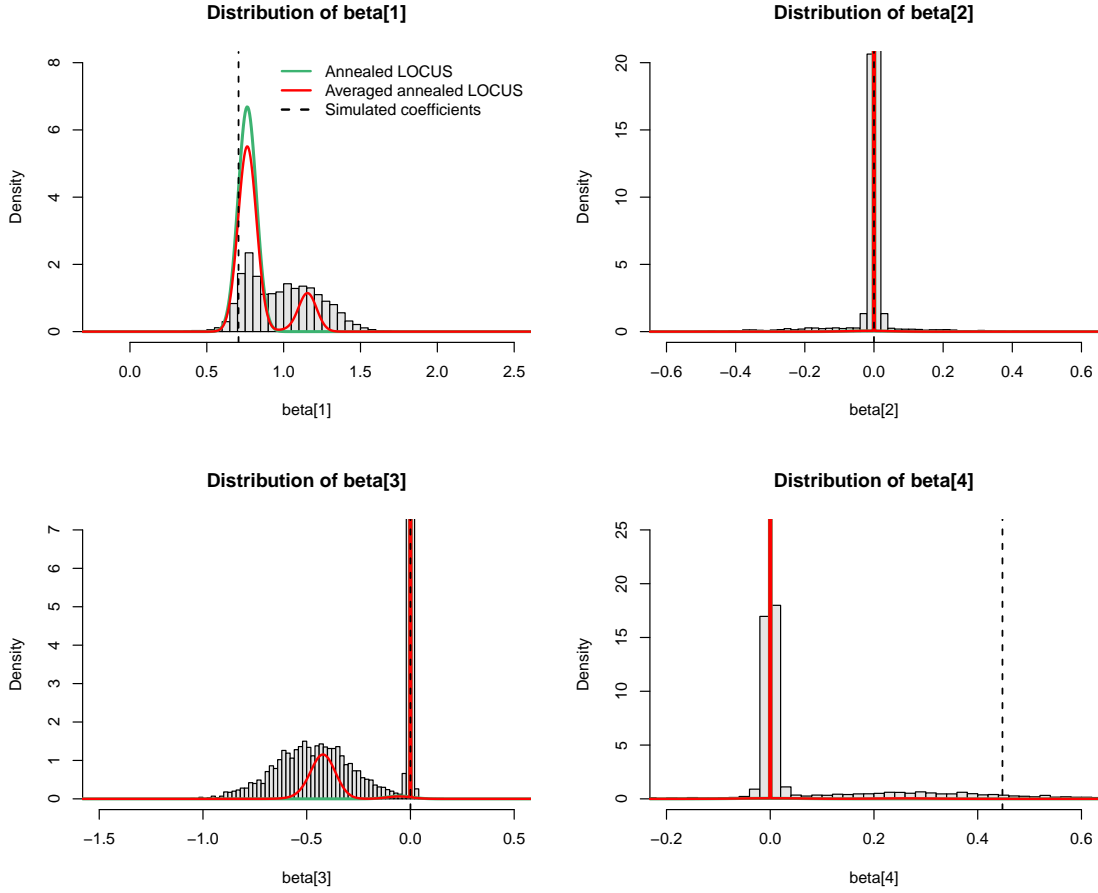


Figure 5.11: Comparison of annealed LOCUS (green) and averaged annealed LOCUS (red) estimated posterior distribution for  $\beta$ , MCMC distributions (histograms)  $\beta$  posteriors as well as the simulated  $\beta$  values (dashed black line). The SNPs are equicorrelated with a correlation coefficient of 0.995.

step added to the LOCUS and averaged LOCUS methods. We have used the same settings than for Figure 5.9, so the histograms and the simulated  $\beta$  values are the same for the two situations. We chose an initial temperature  $T_L = 5$ , and used ten geometric steps.

For all four  $\beta_s$ , annealed LOCUS yields a posterior density that is aligned with averaged annealed LOCUS. The posterior given by annealed LOCUS tends to put mass at the same place than the averaged annealed LOCUS posterior.

As for the non annealing methods, the simulated annealing augmented methods overlap the simulated values for all  $\beta_s$  except for  $\beta_4$  where, the MCMC simulation as well as the augmented methods yield a posterior with values concentrated around zero. As for Figure 5.9, the correlations were too strong and the effect sizes too weak for the algorithms to find a plausible estimation for  $\beta_4$ . And in the same way as in Figure 5.9, the fourth SNP

not being recognised as associated, the algorithms must have compensated on the first SNP, overestimating its effect size  $\beta_1$ .

When comparing the plots of Figures 5.9 and 5.11, one sees that the annealing changed the posterior densities. In Figure 5.9, the posterior density of  $\beta_1$  and  $\beta_3$  were on a wrong mode, but in Figure 5.11 they overlap the simulated  $\beta$ .

## 5.4 Running times

Our method, whether with simulated annealing or not, can be implemented in parallel, which tends to drastically diminish the runtime. Even if the method has to wait until the last run to converge, we would still be quicker than calculating the runs one after the other.

Figure 5.12 shows the running times of the methods, computed on 300 observations of 500 SNPs, 15 SNPs associated with a single trait, and for the averaged versions, 100 different initialisations, measured for 50 replications. For comparison, the runtime of calculating 100 times the LOCUS method before performing a weighted average is shown on the same figure (Serial LOCUS). The advantage of paralleled implementation is highlighted as it takes more than twice as much time to compute 100 iterations of the LOCUS method, one after another, than to compute the averaged LOCUS method.

The averaged annealed LOCUS takes the longest to compute out of the first four methods, as on addition to averaging, the method starts every occurrence with an annealing step. The annealed LOCUS also takes more time to compute than the standard LOCUS method, which confirms the additional time needed for the averaged annealing LOCUS to complete, compared to the averaged LOCUS.

The variability in the runtime of averaged annealed LOCUS is smaller than the variability in the runtime of averaged LOCUS. The annealing step of averaged annealed LOCUS yields the parameters of the different initialisations closer together, as they tend to go in the direction of the global maxima. After the annealing step, the usual algorithm takes a similar amount of time to compute for all the initialisations as the parameters it starts with are closer one from another than they were before the annealing step.

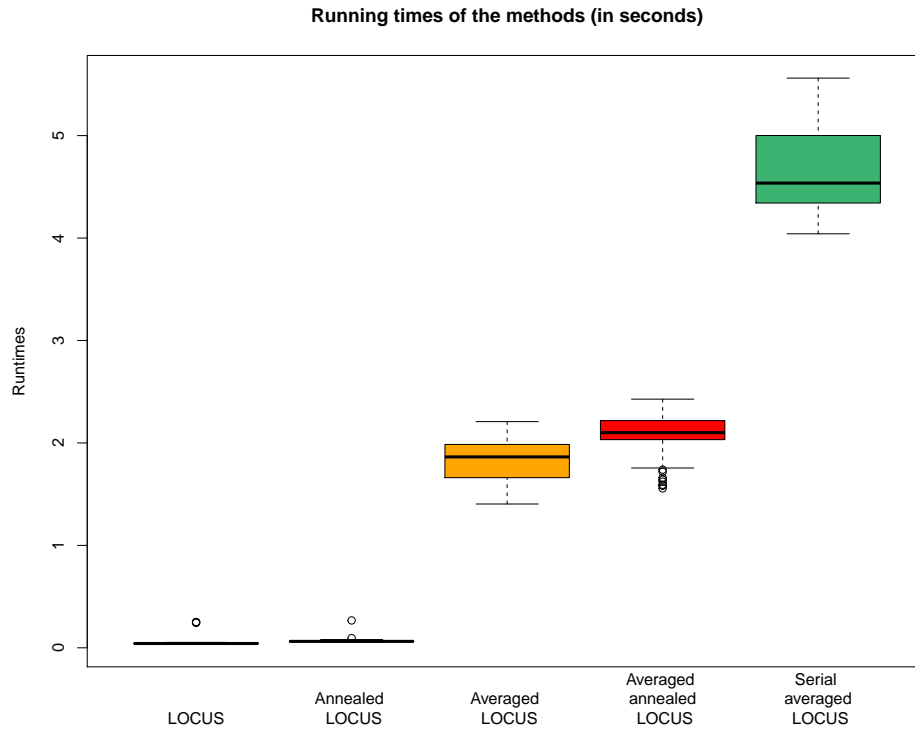


Figure 5.12: Running times, in seconds, of the methods: LOCUS (first), annealed LOCUS (second), averaged LOCUS (third), averaged annealed LOCUS (fourth), and 100 iterations of LOCUS (fifth), computed on 300 observations of 500 SNPs. 15 SNPs are associated with a single trait, and for the averaged versions, 100 different initialisations, averaged over 50 replications. The SNPs are correlated in blocks of ten with an autocorrelation between 0.98 and 0.99. Up to 50% of the response variance is explained by the SNPs. The times have been measured with an Intel Core i5, the methods computed in parallel used 4 cores.



## Chapter 6

# Conclusion

In this work, we proposed a new variational approach based on Bayesian averaging to efficiently deal with strong data correlation in genetic association problems.

We compared the variable selection performance of five methods: the original variational implementation from the package `locus` (LOCUS), our weighted average augmented method (averaged LOCUS), their simulated annealing augmented counterparts (annealed LOCUS and averaged annealed LOCUS), and the weighted average augmented method with all weights equal.

Then, we focused on the posterior distributions of the effect sizes yielded by our four methods, compared them with MCMC inference and the simulated effect sizes. With strong correlations and weak effect sizes, our methods missed an associated SNP, but found it when we weakened the correlations and augmented the effect size of the missed SNP.

We also compared the runtime of the original implementation (LOCUS), our weighted average augmented method (averaged LOCUS), their respective simulated annealing counterparts (annealed LOCUS and averaged annealed LOCUS), and a non parallel implemented weighted averaged method.

Our proposal, averaged LOCUS, outperforms the original LOCUS method in terms of variable selection. On strongly correlated SNPs (0.95 to 0.99), the averaged LOCUS has the best performance, amongst the studied methods. On weaker correlated SNPs (0.5 to 0.95), averaged LOCUS, with or without equal weights, annealed LOCUS, and averaged annealed LOCUS, all have a similar variable selection performance.

Hence, based on the runtime of the different methods, when the correlations between the SNPs is not extremely high, it is quicker to use annealed LOCUS than to use averaged LOCUS, and it has the same variable selection performance. However, on extremely strong correlated data, averaged LOCUS has a better performance than annealed LOCUS and averaged annealed LOCUS, and is quicker to compute than averaged annealed LOCUS.

Several improvements may be considered. First, we also need to evaluate the method performance when considering more than one trait, i.e.,  $q > 1$ . Real eQTL data are made of more than one trait so it would better assess the performance of the methods on relevant data.

Second, we assumed that every model is a priori equiprobable in the averaged LOCUS procedure; other choices could be considered. For example, we could relate the model probability with the expected number of associated SNPs.

Third, for the annealing procedure, we have chosen a geometric schedule, a number of steps  $L$ , and an initial temperature  $T_L$ . It would be good to compare the performance with different choices of initial temperatures, steps, and schedule.

Fourth, we would like to apply this method on real eQTL data.

Finally, provided that all our experiences show good performance, we will optimise our code and integrate it in the `locus` package (<http://github.com/hruffieux/locus>).

# Bibliography

- Altshuler, D., Donnelly, P., and International HapMap Consortium (2005). A haplotype map of the human genome. *Nature*, 437:1299.
- Blei, D. M., Kucukelbir, A., and McAuliffe, J. D. (2017). Variational inference: A review for statisticians. *Journal of the American Statistical Association*, 112:859–877.
- Castillo, I., Schmidt-Hieber, J., and van der Vaart, A. (2015). Bayesian linear regression with sparse priors. *Annals of Statistics*, 43.
- Kullback, S. and Leibler, R. A. (1951). On information and sufficiency. *The Annals of Mathematical Statistics*, 22:79–86.
- Ruffieux, H. (2018a). *echoseq: Replication and simulation of genetic variants, molecular expression levels and other phenotypic data*. R package version 0.2.3.
- Ruffieux, H. (2018b). *Large-scale variational inference for Bayesian joint regression modelling of high-dimensional genetic data*. PhD thesis, EPFL.
- 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.
- Ruffieux, H., Davison, A., Hager, J., Inshaw, J., Fairfax, B., Richardson, S., and Bottolo, L. (2018). A global-local approach for detecting hotspots in multiple-response regression.
- Ruffieux, H., Davison, A. C., Hager, J., and Irincheeva, I. (2017). Efficient inference for genetic association studies with multiple outcomes. *Biostatistics*, 18:618–636.