Averaged variational inference for hierarchical modelling of genomic data

Master thesis

William van Rooij - EPFL

Supervisors: Hélène Ruffieux - Anthony C. Davison

Expert: Eugenia Migliavaccca

14th June 2019

Abstract

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

In this work, we suppose that the data follows a hierarchical regression model linking the genetic variants and outcomes. We are then confronted to 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 big.

Here, we build a method to measure the association between genetic variants and traits based on a method developed by Ruffieux et al. [Ruffieux et al., 2017] that uses variational inference to estimate the posterior distribution of the parameters. We perform a weighted average on the result of multiple different parameter initialisations. We also augment our method with simulated annealing.

We evaluate the performance of our proposal by comparing it to the existing variational approach of Ruffieu et al. [Ruffieux et al., 2017], and also asses its accuracy by comparing it to MCMC inference on a small problem.

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

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 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 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 polymorphisms* (SNPs), 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, of the seventh chromosome located in region ENm014, from Yoruba population [HapMap project Altshuler and Donnelly, 2005]. We can 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 a same block may be, statistically, hard to differentiate. The goal is to represent the probabilities of association between a SNP and a trait, while conveying the uncertainty implied by the block 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 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. The cost of these calculations increases with the number of parameters. Moreover, the higher dimensions, 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

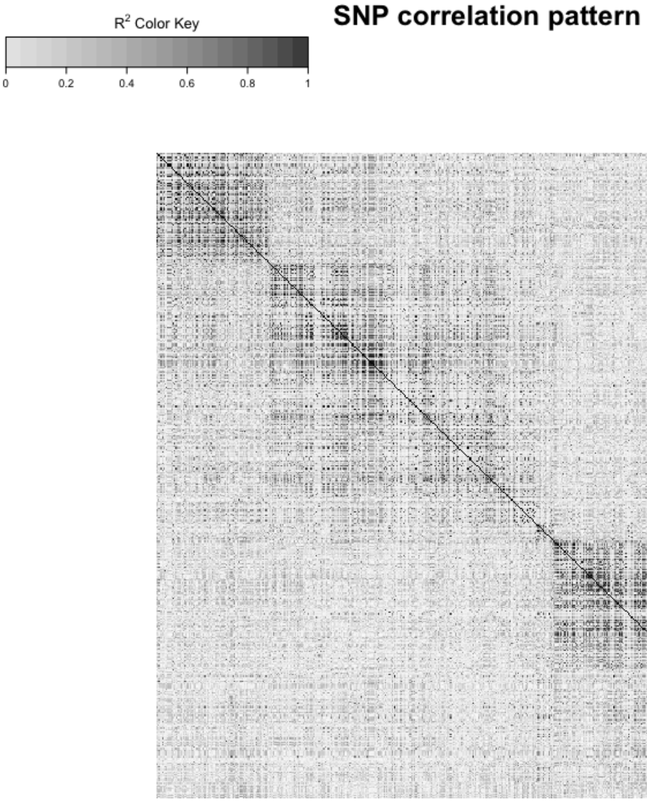

Figure 1.1: Block correlation structure of SNPs taken from Yoruba population HapMap, ENm014 region, chromosome 7 [Altshuler and Donnelly, 2005]. The darker the dot, the stronger the correlation between the two corresponding SNPs.

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

Hierarchical sparse regression for multiple response

2.1 Model

Let $\boldsymbol{X} = (X_1, \ldots, X_p)$ be a centered design matrix, representing the candidate predictor SNPs, and $\boldsymbol{y} = (y_1, \ldots, y_q)$ be a centered response matrix, representing the traits. We consider a hierarchical model, where each response y_t is linearly related with the predictors $\boldsymbol{X}$ 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 $\boldsymbol{\beta}$ is the matrix of regression coefficients. 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 prior distribution of γ_{st} is

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

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

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

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 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, the solution still has one degree of freedom so multiple solutions are possible, e.g.,

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

similarly as in [Ismaël Castillo, 2015].

Parameters σ and ω_s are shared across all the traits, which enables the borrowing of strength across all q 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 $\boldsymbol{\gamma}$ or $\boldsymbol{\beta}$, e.g., for the latter,

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

with

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

where, as mentioned earlier,

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

and δ_0 is the Dirac distribution.

Chapter 3

Variational Inference

3.1 General principles

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 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 $p(\boldsymbol{\theta} \mid \boldsymbol{y})$.

Variational inference minimizes the Kullback–Leibler divergence as a "closeness" measure. Introduced in 1951 by Kullback and Leibler [1951], it 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.,

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

Choosing the family $\mathcal{D}$ can be difficult, as we need it to be simple enough to enable tractable inference, but flexible enough for the approximation $q \in \mathcal{D}$ to be "close" to $p(\boldsymbol{\theta} \mid \boldsymbol{y})$ in terms of the Kullback–Leibler divergence. The approximation will then be

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

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

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

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

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

i.e., 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 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,

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

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

3.2 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 $\boldsymbol{\theta}$, $q \in \mathcal{D}$ 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 $\mathcal{L}(q)$. Hence, the variational family does not directly represent the observed data, they are both linked through the optimisation of the evidence lower bound.

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

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

we keep β_{st} and γ_{st} grouped in order to obtain a "spike-and-slab" for a posteriori for each of the factors, rather than unimodal distributions which would neglect the multimodal behaviour induced by the spike-and-slab prior.

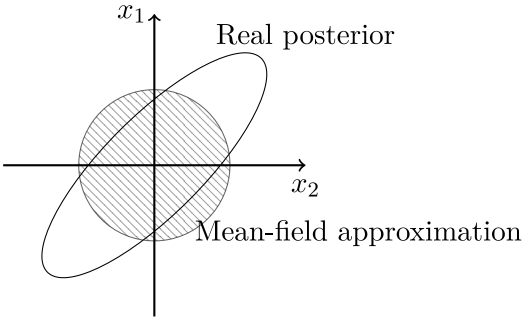

Figure 3.1: To visualise the mean-field approximation, we consider a two dimensional Gaussian distribution, represented in clear in Figure 3.1. The mean-field approximation of the posterior distribution is represented by the barred circle. We see that the mean of the approximation is the same as the 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 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 algorithm

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, optimizing 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:*

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

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 tolerance ε.

Algorithm 1: Coordinate ascent variational inference

```
input     : $p(\boldsymbol{y}, \boldsymbol{\theta})$, 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 $j \in \{1, \ldots, J\}$ do
        set $q_j(\theta_j) \propto \exp\{\mathbb{E}_{-j}[\log p(\theta_j \mid \boldsymbol{\theta}_{-j}, \boldsymbol{y})]\}$
    $\mathcal{L}^{\mathsf{old}}(q) \leftarrow \mathcal{L}(q)$
    $\mathcal{L}(q) \leftarrow \mathbb{E}[\log p(\boldsymbol{\theta}, \boldsymbol{y})] - \mathbb{E}[\log q(\boldsymbol{\theta})]$
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 local optimum thus obtained may depend on the initialization of the $q_j(\theta_j)$, $j = 1, \ldots, J$; different initializations could yield different optima that correspond to different models.

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

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

where $\mu_{\beta,st}$, $\sigma^2_{\beta,st}$, $\gamma^{(1)}_{st}$, a^*_s, b^*_s, η^*_t, κ^*_t, λ^*, and ν^* 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. Suppose $p(\boldsymbol{\theta} \mid \boldsymbol{y})$ is the posterior distribution of $\boldsymbol{\theta} = (\theta_1, \theta_2)$ and one of its 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 and the marginal variances are smaller that those of $p(\boldsymbol{\theta} \mid \boldsymbol{y})$. This results from the optimisation of the reverse Kullback–Leibler divergence

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

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

The lower bound $\mathcal{L}(q)$ tends to be highly multimodal. The ascent algorithm (Algorithm 1) risks to get stuck on 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 one is to introduce a simulated annealing procedure to explore more modes. The second one is to average over multiple parameter initialisations with weights accounting for the likelihood of the obtained mode. We describe these two options next.

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. We next describe how it can be coupled with variational inference to achieve this aim.

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

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

and 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 implies that the approximation will put more mass where the variables best explain the data, and the entropy, that encourages the "dispersion" of the approximation.

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

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

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 disparate approximations

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

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 (4.1) 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 further 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.

The objective for $\mathcal{L}_T(q)$ 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.

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 $\boldsymbol{y}$ may have been obtained from one of multiple models M_k, $k = 1, \ldots, K$, and Δ is the quantity of interest. The posterior distribution:

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

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 $p(\Delta \mid \boldsymbol{y})$ in (4.2), we might be interested in summaries like the posterior mean:

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

The posterior probability for model M_k is given by:

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

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

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

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

and that minimizing the Kullback–Leibler divergence is equivalent to maximizing the evidence lower bound.

Hence, by assuming $\mathcal{L}(q)$ us a tight lower bound on the marginal log likelihood, we can use it as an approximation for $\log p(\boldsymbol{y} \mid M_k)$ 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 our quantity of interest is γ_{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 we described here above. We approximate $\log p(\boldsymbol{y})$ by $\mathcal{L}(q)$ in (4.3), and obtain an approximation for $\mathbb{E}\left[\gamma_{st} \mid \boldsymbol{y}\right]$ considering all the models obtained through the algorithm.

In highly multimodal scenarios, as induced by strong correlation structure, the uncertainty in the selected variables will be conveyed in the resulting approximations for $\mathbb{E}\left[\gamma_s t \mid \boldsymbol{y}\right]$.

To cope with 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.

Chapter 5

Simulations

5.1 Preliminary illustration

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, in order to possibly obtain different optimums. Then we use the evidence lower bound of the different calls as weights to combine the posterior summaries of each initialisation. We will call our method “multiple variational Bayes”, due to the multiple calls of the variational algorithm, and “variational Bayes”, the method consisting just a single call of the algorithm.

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

We first tested our method on simulated data, to be able to compare the results with the truth. We used the `echoseq` `R`-package [Ruffieux, 2018a] to generate blocks of strongly autocorrelated SNPs and traits, as well as associations between them. The SNPs are coded as discrete variables describing their state, we create dependence between them using realisations of multivariate normal variables followed by a quantile thresholding rule.

We have generated 300 observations of 500 SNPs, with latent variable block autocorrelations between 0.95 and 0.99, by blocks of 10 SNPs. For simplicity, we generated just one trait; the extension to multiple traits should produce similar conclusions. We selected five SNPs to be associated with the trait, for better visualisation, all five SNPs are in the 50 first SNPs.

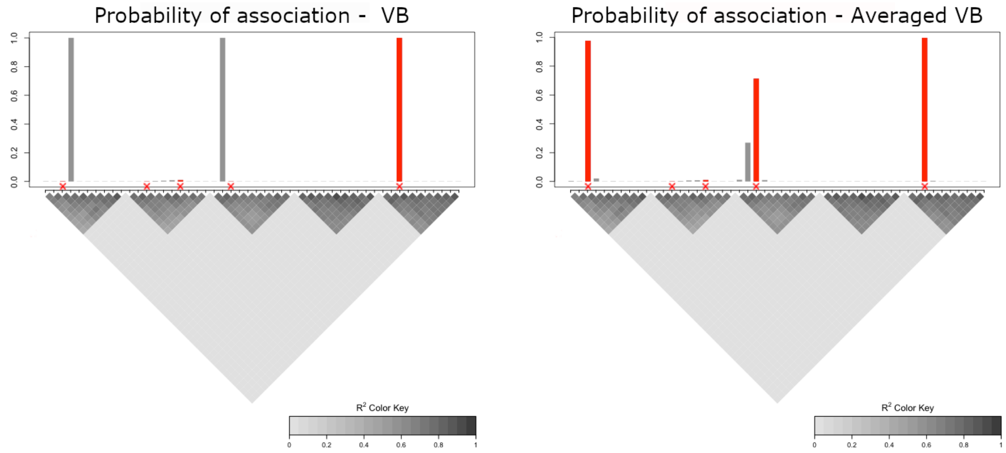

Figure 5.1: Probabilities of association of the 50 first SNPs with a single trait calculated with a single call of the `locus` function (left) and when doing a weighted average on multiple calls of the `locus` function namely, averaged variational inference (right). In red are the five real associated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

Figure 5.1 shows the probabilities of association of the 50 first SNPs, out of 500 used.

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 100 models M_k, $k = 1\ldots,100$. We then calculated

$$\mathbb{E}\left[\gamma_{st} \mid \boldsymbol{y}\right] = \sum_{k=1}^{1} 00\mathbb{E}\left[\gamma_{st} \mid M_k\right] \; p\left(M_k \mid \boldsymbol{y}\right).$$

We see that when using a single call of the `locus` function, 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. The strong correlations can mislead the function into yielding the wrong SNP in the same correlation block.

Our averaged variational inference algorithm does better; it identifies three of the five relevant SNPs.

The two grey peaks of the left plot appear with a lower probability on the right plot; suggesting that the model found by classical variational inference has been considered in the weighted scheme of the averaged variational inference procedure. There seems to be a few other initialization configurations that have also mislead the SNP selection as the second spike of the left plot indicates a nonzero probability of association.

The block wise correlation structure is also better conveyed in the probabilities of association for the averaged variational method. We can see that four SNPs of the middle block have all non null probabilities of association with the trait.

5.2 Variable selection performance

We compared four methods, classical variational inference (LOCUS), averaged variational inference (averaged LOCUS) and their simulated annealing augmented counterparts (annealed LOCUS and averaged annealed LOCUS). We chose four different situations: two of the settings involved 15 associated SNPs (settings A, B), whereas the remaining two had 50 associated SNPs (settings C, D). To ease the computation, we consider only one trait. For a pair of settings, the proportion of the response variance explained by the SNPs could go up to 50% (settings A, C) and, for another pair, up to 80% (settings B, D). The simulated annealing augmented methods have an initial temperature fixed at $T_L = 2$, we have chosen a geometric spacing with ten steps. The sensitivity to these choices could be assessed in dedicated experiments.

Figure 5.2 represent the ROC curves of the four methods, for each of the four settings we mentioned earlier. We truncated the ROC curves as we are interested only in the performance of the methods for small false positive rate. The remaining settings are the same for Figure 5.1. It would be interesting to run other simulations to fully check the variable selection performance of the different methods.

Firstly, we compare LOCUS and averaged LOCUS. We can clearly see in Figure 5.2 that averaged LOCUS's variable selection's performance is better than LOCUS's, as it considers many different initialisations, in our case 100, and attributes to each resulting mode a weight associated to the likelihood of the data being obtained from the corresponding model. We hope that

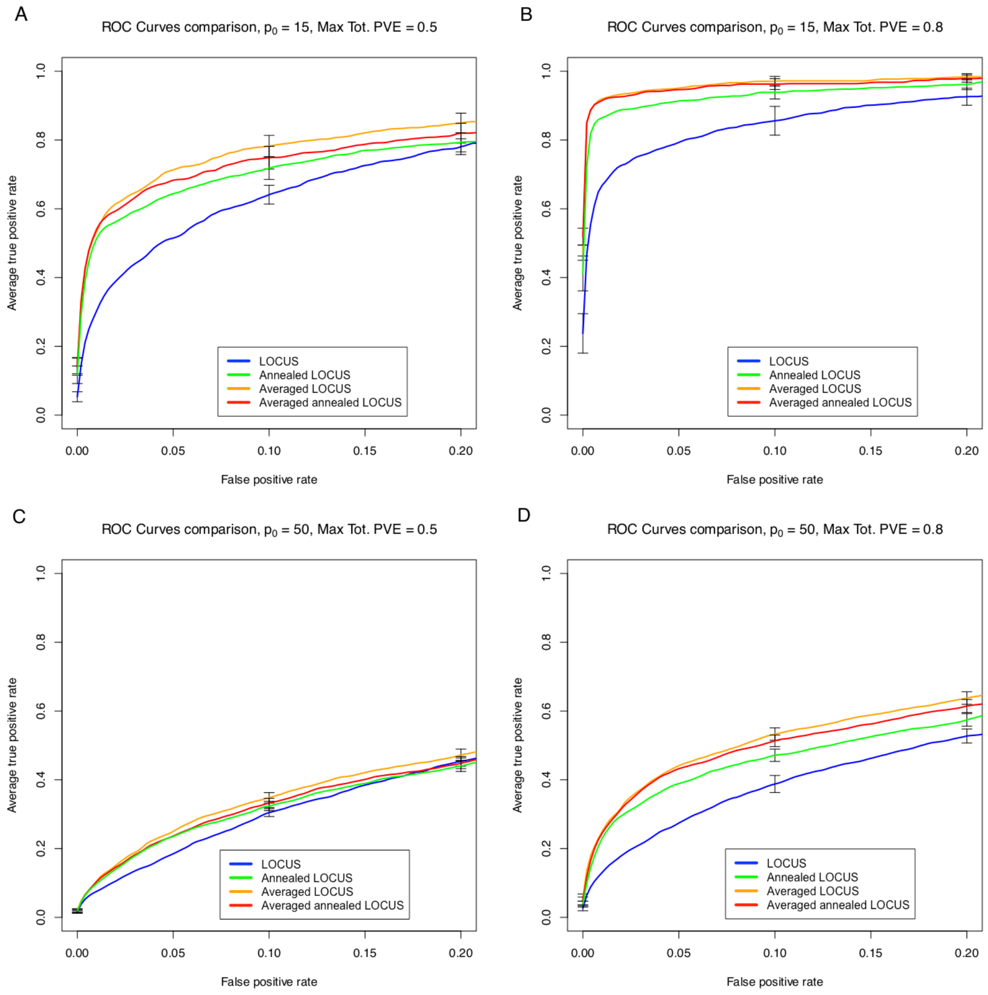

Figure 5.2: Comparison of ROC curves between multiple and single locus, and the same two methods augmented with a simulated annealing step, colored orange, blue, red, and green respectively. Top row: $p_0 = 15$, Left column: Max tot. PVE= 0.5, Bottom row: $p_0 = 50$, Right column: Max tot. PVE= 0.8

the real model can be obtained from one of the obtained modes, then the likelihood of the data originating from said model will be high and the real associated SNPs will be more represented.

Secondly, we can see that when starting both LOCUS and averaged LOCUS with a simulated annealing step, averaged LOCUS remains more powerful than LOCUS, although the improvement is smaller than it is without simulated annealing. This means that the annealing step does not prevent averaged LOCUS's algorithm to end up 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 modes.

Thirdly, annealed LOCUS's variable selection is better than LOCUS's. The simulated annealing step allows the method to reach modes that cannot be reached by the LOCUS method with certain starting parameters.

Fourthly, 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.

Finally, we see that averaged annealed LOCUS's variable selection is very close in performance to averaged LOCUS's: the their confidence intervals overlap. In setting A, the 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 in the wrong models, hence leading the algorithm on the wrong mode.

5.3 Comparison with MCMC inference

Section 5.2 evaluated variable selection performance of the different methods, we now want to compare the accuracy by confronting it with MCMC inference. To do so, we generated some with the `echoseq` `R`-package, and extracted the matrix $\boldsymbol{\beta}$ to have the real parameters. We simulated 300 observations of four SNPs, with a equicorrelated SNPs with correlation coefficient of 0.955. Such a strong correlation level can mislead the methods in the selection of the associated SNPs.

We compare the posterior distributions of $\beta_1, \ldots, \beta_4$ 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 burn the first half.

For the same reasons, we consider a very small problem 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 $\beta_1, \beta_4 \neq 0$.

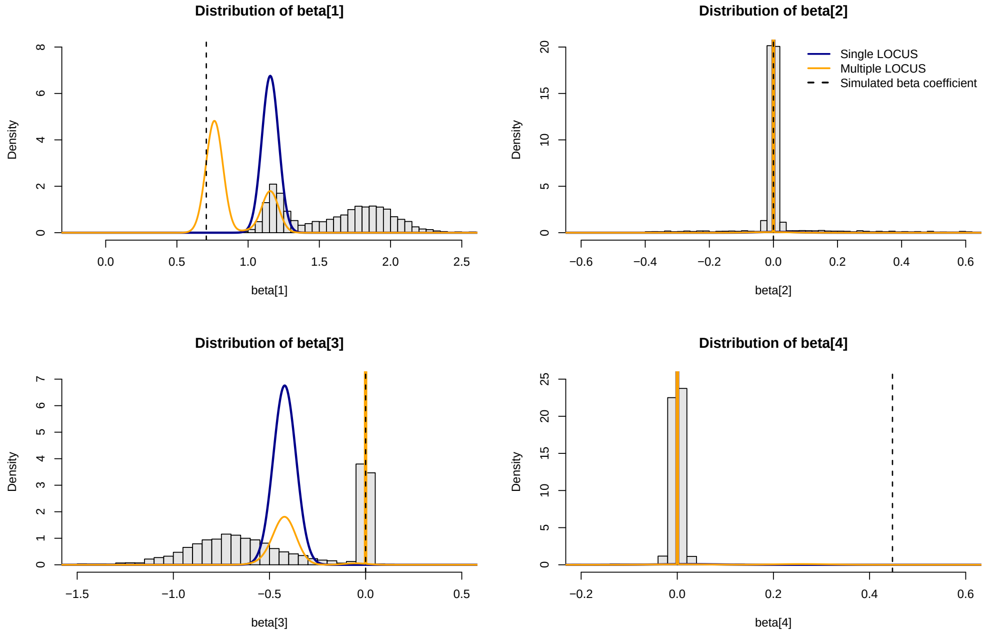

Figure 5.3: Comparison of LOCUS (blue) and averaged LOCUS (orange) calculated posteriors for $\boldsymbol{\beta}$, MCMC simulated (histograms) $\boldsymbol{\beta}$ posteriors as well as the real (dashed black line) $\boldsymbol{\beta}$ values.

In Figure 5.3, we have plotted LOCUS and averaged 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 superimposed.

Firstly, we can see that averaged LOCUS puts mass near the simulated values of β_s for every β_s but for β_4, where it finds the same estimation as the MCMC inference and the LOCUS methods. However, LOCUS only agreed to some extent with the MCMC distribution only for β_2. This confirms what we read in the ROC curves of Figure 5.2, where we saw that the performance of the averaged LOCUS is better that the performance of LOCUS.

Secondly, when LOCUS and averaged LOCUS do not yield the same value for the parameters, the result of LOCUS is visible in the distribution of averaged LOCUS. This is given by the fact that averaged LOCUS considers the mode obtained from LOCUS in its averaging, and in that case, the mode obtained from LOCUS was relevant. This can be read in Figure 5.1, where we can see that LOCUS selects a wrong SNP and that even if averaged LOCUS selects the right SNP, we can still see in the probabilities of association it calculated, the SNP selected by LOCUS.

Finally, β_4 is supposed to be non null, but the MCMC simulations and the approximations given by LOCUS and averaged LOCUS methods are all null. The strong correlation gave the wrong mode too much weight, giving the illusion that it was the global 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 or not.

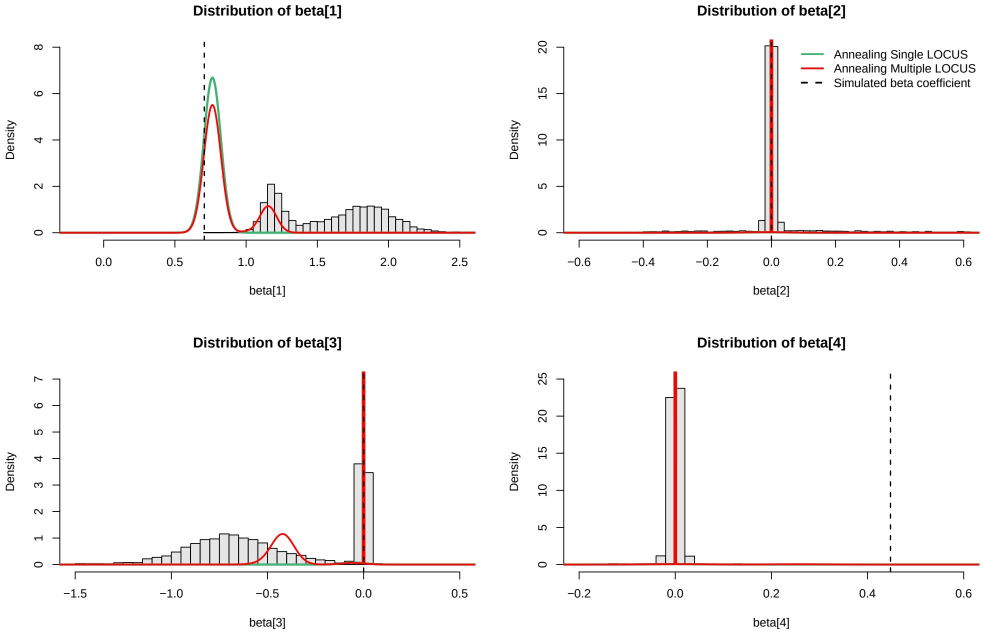

Figure 5.4: Comparison of annealed LOCUS (green) and averaged annealed LOCUS (red) calculated posteriors for $\boldsymbol{\beta}$, MCMC simulated (histograms) $\boldsymbol{\beta}$ posteriors as well as the real (dashed black line) $\boldsymbol{\beta}$ values.

Figure 5.4 shows the same posteriors as Figure 5.3, but with a simulated annealing step added to LOCUS and averaged 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 chose an initial temperature $T_L = 5$, and used ten geometric steps.

We can now see that for all four β_s, LOCUS yields a posterior density that is more aligned with averaged LOCUS. The posterior given by LOCUS tends to put mass at the same place than the averaged LOCUS posterior.

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

However, there exists an initialisation of the parameters for annealed LOCUS that yields a posterior not aligned on the real values of β_1 and β_3. The posterior distribution of averaged annealed LOCUS puts mass where the shown posterior of annealed LOCUS has none.

When comparing the plots of Figures 5.3 and 5.4, the annealing changed the density of the posterior yielded by the 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 overlap the simulated $\boldsymbol{\beta}$.

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

5.4 Running times

Our method, whether with simulated annealing or not, can be implemented in parallel, which can drastically diminish the runtime. 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.

Figure 5.5 shows the running times of the four methods worked on so far. Average LOCUS takes twice the time of LOCUS to compute, due to the first annealing steps. The two averaged methods take more time than the two others, which is expected, but knowing they each are made of 100 initialisations highlights the efficiency of the parallel implementation.

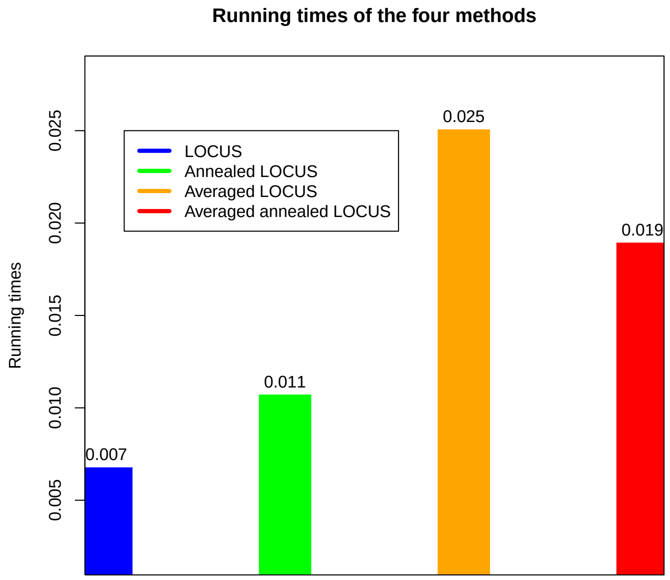

Figure 5.5: Running times of the four methods: LOCUS (blue), annealed LOCUS (green), averaged LOCUS (orange), and averaged annealed LOCUS (red), computed on 500 SNPs, a single trait, 100 different initialisations, and averaged over 20 iterations.

Chapter 6

Conclusion

We compared the variable selection performance, posterior distributions, and runtime of four methods: the original variational implementation from the package `locus` (LOCUS), our weighted average augmented method (averaged LOCUS), and their simulated annealing augmented counterparts (annealed LOCUS and averaged annealed LOCUS).

The averaged 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 performance of averaged LOCUS is better than the performance of LOCUS, but when augmenting the methods with annealing, the gap gets smaller.

The annealed LOCUS method performs better than the LOCUS method, but the averaged annealed LOCUS has approximately the same performance than the averaged LOCUS method.

The runtime of LOCUS is a lot smaller than the runtime of averaged LOCUS but an important point for averaged LOCUS 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.

With more time, we would compute the methods performances when considering more that one trait i.e., $q > 1$. The real data are made of more than one trait so it would better represent the performances of the methods on relevant data.

We consider that every model is equiprobable in the averaged LOCUS when we could choose the probabilities to be variable. For example, we could relate the model probability with the number of expected associated SNPS.

We have chosen a geometric schedule for the annealing parts for our performance analysis. In Section 4.2, we have defined three schedules, and we have chosen a certain amount of steps L and an initial temperature T_L. We could find the best combination of temperature, steps, and schedule, such that the performance is the best.

To be able to tell if our algorithm adequately explores the local modes, we want to represent the posterior of $\boldsymbol{\beta}$ as well as its initial and final state similarly as V. Rockova [Rocková, 2017] did.

We will optimise the code that we implemented, to have a comparison with the other methods commonly used. We may include the function in H. Ruffieux's `R`-package (http://github.com/hruffieux/locus).

Finally, we would like to apply this method on real-life data.

Bibliography

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). Variational inference: A review for statisticians.

Ismaël Castillo, Johannes Schmidt-Hieber, A. v. d. V. (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.

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

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 Bayesian Inference for Genetic Association withMultiple Outcomes.* 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. C., Hager, J., and Irincheeva, I. (2017). Efficient inference for genetic association studies with multiple outcomes.