atlasqtl: an R package for variable selection in sparse regression with hierarchically-related outcomes

Hélène Ruffieux

06/15/2022

Statistical problems in the high-dimensional setting multiply, prompted by the proliferation of technologies capable of measuring large volumes of information, whether on health and lifestyle parameters, molecular entities or even galaxies. As well as growing in dimension, the datasets collected are also growing in complexity, which calls for elaborate and flexible modelling approaches. When coupled with robust and efficient inference, Bayesian hierarchical modelling is a powerful framework for describing intricate dependencies at the scale required by current applications, while conveying uncertainty in a coherent fashion. In this note, we present the R package `atlasqtl`, which implements a scalable hierarchical framework for variable selection in regression problems with high-dimensional predictor and response spaces.

Model and inference

The model consists of series of hierarchically-related spike-and-slab regressions that permit borrowing information across large numbers of responses. A graphical representation is provided in Figure 1, and the full model and inference algorithm are detailed in Ruffieux et al. (2020). Each pair of predictor X_s and response y_t has its corresponding regression coefficient β_{st} and spike-and-slab binary latent variable γ_{st}, from which posterior probabilities of association are conveniently obtained, $\mathrm{pr}(\gamma_{st} = 1 \mid y)$, and employed to implement Bayesian false discovery rate control. The model is also tailored to the detection of *hotspots*, namely, predictors associated with several responses: the top-level hierarchy entails a probit submodel which involves a response-specific contribution to the spike-and-slab probability of association, via ζ_t, and a predictor-specific modulation of this contribution, via θ_s. The latter parameter also acts as the propensity of predictor X_s to be a hotspot and is assigned a horseshoe prior, which adapts to the overall problem sparsity (via the global scale σ_0), while flexibly capturing hotspot effects (via the Cauchy tail of the local scale λ_s). Joint inference requires special attention as the binary latent matrix $\Gamma = \{\gamma_{st}\}$ creates a discrete search space of dimension $2^{p\times q}$. To overcome this complication, `atlasqtl` implements a variational inference algorithm based on a structured mean-field factorisation and efficient batch updates. The algorithm is augmented with a simulated annealing scheme that improves the exploration of highly multimodal spaces by introducing so-called "temperature" parameters controlling the degree of separation of the modes of a series of "heated" distributions (Ruffieux et al. (2020)).

`atlasqtl` can be employed in any sparse multiple-response regression setting. Hereafter we discuss a use case in the context of expression quantitative trait locus (eQTL) analysis, in which *hotspot genetic variants*, controlling many molecular traits at once, may be responsible for important functional mechanisms underlying specific disease endpoints — such studies aim to clarify the genetic architecture of diseases by estimating associations between up to a few million candidate predictors (genetic variants) and several thousand responses (molecular traits, such as genomic, proteomic, metabolomic levels).

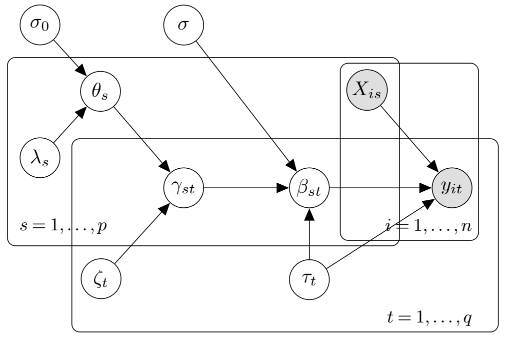


Figure 1: Graphical representation of the model. The shaded nodes are observed, the others are inferred.

Installation

The `atlasqtl` package is written in R with C++ subroutines. It can be installed following the instructions provided at https://github.com/hruffieux/atlasqtl#installation. Note that the GSL library must be installed prior to installing the package.

Data and reproducibility

The material used in this example is available at https://github.com/hruffieux/software_corner_ibs_bulletin for replication purposes. It involves synthetic genotyping data generated using our in-house R package `echoseq` available at https://github.com/hruffieux/echoseq; these data emulate real genotypes (single nucleotide polymorphisms, "SNPs") in the *LYZ* gene region, which cannot be share without data access agreement. The expression data consist of monocyte levels before and after immune stimulation (resting, IFNg, LPS2h, LPS24h), as well as B-cell levels. To each of these transcriptomic datasets corresponds a separate eQTL problem.

Analysis

We next illustrate the main package functionalities on five different eQTL analyses (whose names, resting monocytes, IFNg, LPS2h, LPS24h monocytes and B-cells, are gathered in the vector `vec_type`, and which can be run in parallel = here, using `n_cpus = 4` cpus). The corresponding datasets are stored in the list `list_data`.

`atlasqtl` requires the specification of just two hyperparameters via the elicitation of a prior mean `mu_t` and variance `v_t` for the number of SNPs associated with each trait. The sensitivity of inference to these choices is typically limited, and essentially non existent once a permutation-based FDR threshold is employed (where the "null-case" permutation runs use the same prior specifications).

The remaining model parameters are inferred by variational inference; in particular, the hotspot propensity is assigned a horseshoe prior which circumvent ad-hoc specifications of top-level variances (a case where inference is prone to strong biases). This specification also has desirable multiplicity adjustment properties for dealing with the large-response case, see Ruffieux et al. (2020).

The next chunk runs the five ATLASQTL analyses in parallel, also storing the overall run time. We are using the default annealing setting, namely a geometric schedule on the inverse temperature, with initial temperature of 2 and 10 different temperatures. This specification, as well as alternative options, are described in greater details on the help page of the function, by running `?atlasqtl`. A number of additional settings (e.g., tolerance, maximum number of iterations, checkpointing, etc.) are also detailed there.

Note that the parallel execution relies on the R package `parallel` which has already been installed as part of the `atlasqtl` installation.

```
require(atlasqtl)

mu_t <- 1
v_t <- 4

n_cpus <- 4 # Here we use the $4$ cores of our laptop, but pl
                                                                          us <- 1` co

list_out <- parallel::mclapply(vec_type, function(type) {

  snps <- list_data[[type]]$snps
  expr <- list_data[[type]]$expr

  stopifnot(rownames(snps) == rownames(expr))

  atlasqtl(Y = expr, X = snps,
           p0 = c(mu_t, v_t),
           add_collinear_back = TRUE)

}, mc.cores = n_cpus)

names(list_out) <- vec_type
```

The object returned by `atlasqtl` contains a range of useful posterior quantities, which can be employed to assess:

- the pairwise associations between each pair of SNP and trait: using the variational posterior probabilities (PPIs) stored in `gam_vb` ($p \times q$ matrix) and the variational posterior means of the regression estimates stored in `beta_vb` ($p \times q$ matrix);
- the hotspot propensities: using the variational posterior mean of θ_s stored in `theta_vb` (vector of length p).

It also contains diagnostic values on the final status of convergence and number of iterations used for the coordinate ascent variational algorithm.

We print here a snapshot of these quantities for the first unstimulated-monocyte eQTL analysis.

```
atlasqtl_unstim <- list_out$unstim
summary(as.vector(atlasqtl_unstim$gam_vb)) # PPIs, pr(gamma = 1 | y)
```

```
##      Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
## 0.0001882 0.0002576 0.0003345 0.0037925 0.0007167 1.0000000
```

```
summary(as.vector(atlasqtl_unstim$beta_vb)) # E(beta | y)
```

```
##       Min.    1st Qu.     Median       Mean    3rd Qu.       Max.
## -0.5867404 -0.0000046  0.0000000  0.0000206  0.0000046  0.6948798
```

```
summary(atlasqtl_unstim$theta_vb) # E(theta | y)
```

```
##       Min.    1st Qu.     Median       Mean    3rd Qu.       Max.
## -0.0003263 -0.0003076 -0.0002945  0.0403931 -0.0002676  1.6039678
```

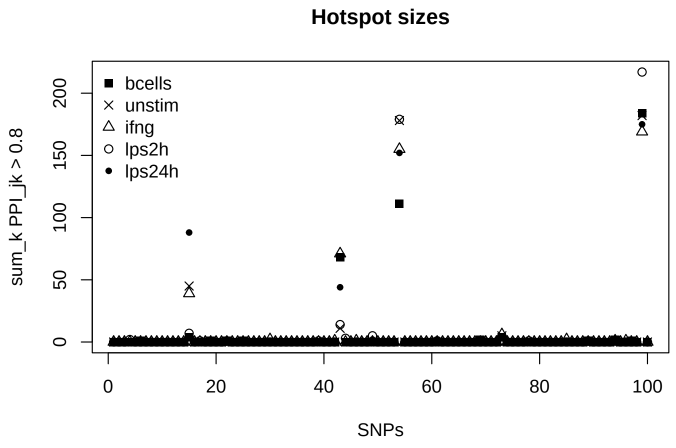
Finally, we display a Manhattan-type plot which indicates the position of the hotspots, as estimated in the five different eQTL analyses.

Also note that permutation analyses can be run to compute calibrated Bayesian FDR thresholds on the PPIs; we don't implement this for computational economy and use an arbitrary threshold instead. The procedure to derive these permutation-based FDR threshold is described in details in Ruffieux et al. (2017) and Ruffieux et al. (2020).

```
thres <- 0.8
list_rs_thres <- lapply(list_out,
                        function(ll_type)
                          rowSums(ll_type$gam_vb>thres))

vec_pch <- c(15, 4, 2, 21, 20)

plot(list_rs_thres[[1]], pch = vec_pch[1],
     ylim = c(0, max(unlist(list_rs_thres))),
     main = "Hotspot sizes",
     xlab = "SNPs",
     ylab = paste0("sum_k PPI_jk > ", thres))
for (type_id in 2:5) {
  points(list_rs_thres[[type_id]], pch = vec_pch[type_id])
}
legend("topleft", legend = vec_type, pch = vec_pch,
       bty = "n")
```



References

Ruffieux, H., A. C. Davison, J. Hager, J. Inshaw, B. Fairfax, S. Richardson, and L. Bottolo. 2020. "A Global-Local Approach for Detecting Hotspots in Multiple Response Regression." *The Annals of Applied Statistics* 14: 905–28.

Ruffieux, H., A. C. Davison, J. Hager, and I. Irincheeva. 2017. "Efficient Inference for Genetic Association Studies with Multiple Outcomes." *Biostatistics* 18: 618–36.