BayesRCO: extending BayesRC to overlapping annotations

May 25, 2021

[∗,1]Fanny Mollandin [∗,†]Andrea Rau [∗]Pascal Croiseau
[∗]Université Paris-Saclay, INRAE, AgroParisTech, GABI [†]BioEcoAgro Joint Research Unit, INRAE, Université de Liège, Université de Lille, Université de Picardie Jules Verne

1 Overview

The BayesRCO software includes five different Bayesian genomic prediction models: BayesCπ [1], BayesR[2], BayesRC[3], BayesRCπ, and BayesRC+. These five models are Bayesian Gaussian mixture models for genomic prediction of complex traits, based on Markov Chain Monte Carlo (MCMC) algorithms. These methods also allow the study of the underlying genomic architecture of these traits, and potentially QTL mapping. Moreover, the authors propose here two new methods, BayesRC+ and Bayesπ, to integrate multiple overlapping functional annotations. This document is intended to help in the use of this software, and the possible options.

DISCLAIMER: This software is based on an older version of BayesR written by Gerard Moser and available here: https://github.com/syntheke/bayesR; therefore there are many similarities between this document and the user manual proposed by Moser.

2 Models

2.1 Distribution

All five models exploit the same prediction equation, and seek to best estimate the vector of phenotypes y by best estimating the vector of SNP β effects:

$$\begin{aligned} \mathbf{y} &= \mu\mathbb{1}_n + \mathbf{X}\boldsymbol{\beta} + \mathbf{e}, \\ \mathbf{e} &\sim N(0, \mathbf{I}_n\sigma_e^2) \end{aligned} \tag{1}$$

We differentiate these models by the distribution attributed *a priori* to β, as indicated in the table below. In each model, SNPs can be assigned to different effect sizes, 2 (null and non-null) for bayesCπ, or 4 (null, low, medium and high) for the other methods. In addition, BayesRC, BayesRCπ and BayesRC+ exploit functional information, separating SNPs into several categories.

Method	SNP effect prior distribution	Size effect classes	Annotations
BayesCπ	$\beta_i \sim \pi\mathcal{N}(0,0) + (1-\pi)\mathcal{N}(0,\sigma_\beta^2)$	2	No
BayesR	$\beta_i \sim \sum_{K=1}^4 \pi_K \mathcal{N}(0, k\sigma_g^2)$	4	No
BayesRC	$\beta_i \vert a = A(i) \sim \sum_{K=1}^4 \pi_{K,a}\mathcal{N}(0, k\sigma_g^2)$	4	Yes, disjointed
BayesRC+	$\beta_i \vert a \in \mathbf{A}(i) \sim \sum_{a\in\mathbf{A}(i)} \sum_{K=1}^4 \pi_{K,a} N(0, k\sigma_g^2)$	4	Yes, overlapping
BayesRCπ	$\beta_i \vert a \in \mathbf{A}(i) \sim \sum_{a\in\mathbf{A}(i)} p_{i,a} \sum_{K=1}^4 \pi_{K,a} N(0, k\sigma_g^2)$	4	Yes, overlapping

with σ_g^2 the total additive genetic variance, $\pi = (\pi_1, \pi_2, \pi_3, \pi_4)$ the mixing proportions such that $\sum_{i=1}^4 \pi_i = 1$, $p_{i,a}$ the mixing proportions of the SNP i in its set of annotations $\mathbf{A}(i)$ such that $\sum_{a\in\mathbf{A}(i)} p_{a,i} = 1$, and $k = \{0, 0.0001, 0.001, 0.01\}$.

2.2 BayesRCπ and BayesRC+

We have developed BayesRCπ and BayesRC+ to be able to extend BayesRC to use overlapping functional annotation data. The first model is a mixture model, which assigns SNPs to one of their annotations at each iteration, based on likelihood. This step is analogous to the step of assigning SNPs to one of four SNP effect classes. The second model is additive, a SNP can cumulate effects drawns from different annotations.

[On parle des situations dans lesquels ils sont biens ? Mais comme on sait pas trop pour l'instant héhé]

2.3 Algorithms

As exact computation of the posterior distribution is intractable for this models, Bayesian inference is performed by obtaining draws of the posterior using a Gibbs sampler. Model parameters are estimated using the posterior mean across iterations, after excluding the burn-in phase and thinning draws. By default, the Gibbs sampler runs for a total of 50,000 iterations, including 20,000 as a burn-in and a thinning rate of 10.

3 Download & Compilation

We started from the source code of BayesR proposed by Moser (available here https://github.com/syntheke/bayesR/), and thus kept the same file structure:
- RandomDistributions.f90 : auxiliary file containing various random generator [What we added from Moser?]
- baymodsRCO.f90 : support module for BayesRCO containing commun variables and routine [COPIER COLLER DE MOSER...]
- bayesRCO.f90 : main program

The program can be compiled with a FORTRAN95 compiler on a Unix operating system using the following commands.

```
gfortran RandomDistributions.f90 baymodsRCO.f90 bayesRCO.f90 –o bayesRCO
```

j'ai pas mis les autre compilateur ifort, grave ?

4 Inputs

4.1 Data

The program requires PLINK binary ped file format. It requires '*.bim and *.fam files to determine the number of SNPs and the number of individuals and a '*.bed' file for the genotype information.

Genotype data The program requires genotypes in PLINK binary format in default-SNP major mode. Since BayesR includes all genotypes in the model, samples missing a genotype call cannot simply be omitted. Missing genotypes are replaced by the mean genotype value of a given marker.

Phenotype data The program reads "column 6" as the phenotype column from the PLINK *.fam file. A different phenotype column can be specified by using the "–n [num] "option, where "–n 1" uses the original 6th column (default), "–n 2" uses column 7 and so forth. Missing phenotypes (or phenotypes to be predicted) must be coded as 'NA'.

4.2 Functional annotations

We can represent SNP annotation categories as a binary design matrix, with SNPs in lines and annotations in columns. We differentiate two types of annotation matrix, non-overlapping (for BayesRC) or overlapping (for BayesRC+ or BayesRCπ) :

$\begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ . & . & . \\ 1 & 0 & 0 \\ 1 & 0 & 0 \end{pmatrix}$ is a **non-overlapping annotation** matrix, all the SNPs must be assigned one and only annotation.

$\begin{pmatrix}0 & 1 & 1\\ 0 & 1 & 0\\ . & . & .\\ 1 & 0 & 0\\ 1 & 0 & 1\end{pmatrix}$ is an **overlapping** annotation matrix, all the SNPs must be assigned in at least one annotation.
We read it as: the first SNP belongs to the annotations 2 and 3, the second to the annotation 2, and so on.

As recommanded in MacLeod paper[3], it is important to have big enough annotation (above 1000 SNPs) to avoid estimation problem of the π_a parameters.

4.3 General Inputs

Input	Function	default
-bfile	prefix PLINK binary files	None
-out	prefix for output	None
-n	phenotype column	1
-vara	SNP variance prior	0.01
-vare	error variance prior	0.01
-dfvara	degrees of freedom Va	-2.0
-dfvare	degrees of freedom Ve	-2.0
-delta	prior for Dirichlet	1.0
-msize	number of SNPs in reduced update	0
-mrep	number of full cycles in reduced update	5000
-numit	length of MCMC chain	50000
-burnin	burnin steps	20000
-thin	thinning rate	10
-ndist	number of mixture distributions	4
-gpin	effect sizes of mixtures (% x Va)	0.0,0.0001,0.001,0.01
-seed	initial value for random number	0
-predict	perform prediction	f
-snpout	output detailed SNP info	f
-cat	output SNP categories per iteration	None
-beta	output SNP effect per iteration	None
-permute	permute order of SNP	f
-model	model summary file (for prediction)	None
-freq	SNP frequency file (for prediction)	None
-param	SNP effect file (for prediction)	None
-ncat	number of SNP annotations	1
-catfile	SNP annotation matrix file	None
-additive	run BayesRC+	f
-bayesCpi	run BayesCπ	f

4.4 Run the software

We use the software in two steps, a first one for the training, thus the estimation of the model parameters, and a second one for the prediction. We use two separate datasets, one with phenotype values, and one without. We show here how to launch these two features. Be careful, the SNPs must match between the two datasets. By default, the software proceeds to a BayesRCπ.

4.4.1 BayesRCπ

```
path/bayesRCO -bfile [prefix_learning] -out [prefix_learning] -ncat [number of annotations] -catfile [annotation_matrix]
path/bayesRCO -bfile [prefix_validation] -out [prefix_validation] -model [prefix_learning].model -freq [prefix_learning].frq -param [prefix_learning].param -ncat [number of annotations] -catfile [annotation_matrix]
```

4.4.2 BayesRC+

To run BayesRC+, please use the flag -additive in the training part.

4.4.3 BayesRC

As BayesRC can be considered as a particular case of BayesRCπ or BayesRC+, you can simply run one of the two previous methods with a non overlapping annotation matrix.

4.4.4 BayesR

BayesR can be seen as a BayesRC with one annotation, so BayesR can be run the same way that BayesRC, with a annotation matrix being a vector of 1, the same length that the number of SNPs as: $\begin{pmatrix} 1 \\ 1 \\ . \\ 1 \\ 1 \end{pmatrix}$. As the default number of ncat is 1, there is no need to specify this option/

4.4.5 BayesCπ

Lastly, BayesCπ can be run similarly to BayesR, by adding the flag -bayesCpi.

4.4.6 Options

A priori information for variance components
Prior inverted-chi squared distribution can be specified for both additive and residuals variance (σ_a^2 and ${}_e^2$). Scale and degrees of freedom (df) for the variance components are required. “Flat” (improper) distributions can be specified by setting df to -2. It is also possible to specify the heritability of the trait by setting dfvara to -3.0 (i.e. -dfvara -3.0). In this case the scale parameter is treated as the heritability and the SNP-based variance is set (fixed) to σ_a^2=heritability$\times\sigma_y^2$ (σ_y^2 being the phenotypic variance).
Effect size Dirichlet prior (all)
The default is to use a uniform and almost uninformative prior for the mixture distribution with a pseudo-observation of 1 (SNP) for each class. Different priors can be specified using the “delta [num]” option. For example, “-delta 3,2,1” specifies a prior with 3, 2 and 1 pseudoobservations for classes 1 to 3 of a 3-component mixture model, “-delta 2” sets the prior to 2 for all mixture components.
Annotation Dirichlet prior (BayesRCπ)
For the moment there is no parameter to change the value of the annotation assignment *prior*. It is likely to be possible to modify the value of this priority in a future version. **Mixture model**
The BayesR, BayesRC, BayesRC+ and BayesRCπ models assume that the true SNP effect is derived from a series of normal distributions. The default models uses 4 mixture distributions with SNP variances of 0, 0.0001, 0.001 and 0.01, so that the variance (S) of the jth SNP has 4 possible values: S1=0, S2=0.0001$\times\sigma_a^2$, S3=0.001$\times\sigma_a^2$, S4=0.01$\times\sigma_a^2$. Different mixture models can be specified using the “–ndist [num]“ and “–gpin [num]“ options. For example, “–ndist 3 –gpin 0.0,0.001,0.05“ fits a 3 component mixture with SNP variances S1=0, S2=0.001$\times\sigma_a^2$, S3=0.05$\times\sigma_a^2$.
MCMC sampling
The default is to use a chain length of 50,000 samples (“–numit”) with the first 20,000 samples (“–burnin”) being discarded, and using every 10th sample (“–thin”) for posterior inference. To improve mixing, one can use the option “–permute” to update SNP effects in random order.

5 Outputs

The outputs all have the same name as specified when launching the software, followed by a suffix corresponding to their type, as so:

5.1 Frequency file

name_output.type
One column containing the SNP allele 2 frequency.

5.2 Log File

The file name prefix is as specified by "–out [prefix_training]". The suffix '.log' is appended to give the file name. This is a descriptive file and provides a summary of the run parameters used and the number of records processed.

5.3 Frequency File

Contains allele frequency of the '2' allele. The suffix '.frq' is appended to the prefix. This file is required for scaling and centering genotypes for prediction analysis. The SNP order has to be the same as the genotype input file.

5.4 Model File

The suffix 'model' is appended to the output prefix. This file contains means of the posterior samples of model parameters:
Mean: intercept
Nsnp: number of SNPs in model Va: genetic variance explained by SNPs
Ve: residual variance
Nk1_1,...,Nkk_j: number of SNPs in mixture components 1 to k and annotation 1 to j
Pk1_1,...,Pkk_j: proportion of SNPs in mixture component 1 to k and annotation 1 to j
Vk1_1,...,Vkk_j: sum of squares of SNP effects in mixture component 1 to k and annotation 1 to j

5.5 Hyperparameters file

File 'prefix.hyp' gives posterior parameter estimates for each MCMC sample:
Replicate: iteration number
Nsnp: number of SNPs in model
Va: genetic variance explained by SNPs
Ve: residual variance
Nk1_1,...,Nkk_j: number of SNPs in mixture components 1 to k and annotation 1 to j
Pk1_1,...,Pkk_j: proportion of SNPs in mixture component 1 to k and annotation 1 to j
Vk1_1,...,Vkk_j: sum of squares of SNP effects in mixture component 1 to k and annotation 1 to j

5.6 Parameters file

The suffix 'param' is appended to the output prefix. The SNP order is the same as the genotype input file. This file contains mean posterior estimates for each individual SNP:
PIP_1, ..., PIP_k: Posterior inclusion probabilities of the SNP in mixture classes 1 to k
beta: posterior SNP effect
PAIP_1, ..., PAIP_j : Posterior annotation inclusion probabilities of the SNP in annotation 1 to j (useful for BayesRCπ, otherwise gives the annotations each SNP belong to).
Vbeta: variance of the posterior SNP effects across iterations
Vi : posterior variance of the SNP effects

5.7 Genetic value file

This file outputs the predicted genomic values (GVs). The output prefix is used to give the file name 'prefix.gv'.

5.8 Optional files

-snpout Output is in sparse format: 'mixture class:SNP:effect size'. The SNP number (SNP) corresponds to the row number of the SNP in the PLINK "bim" file.
-cat output SNP categories per iteration
-beta output SNP effect per iteration

References

[1] Habier D, Fernando RL, Kizilkaya K, Garrick DJ. Extension of the bayesian alphabet for genomic selection. BMC Bioinformatics. 2011 Dec;12(1):186. Available from: https://bmcbioinformatics.biomedcentral.com/articles/10.1186/1471-2105-12-186.

[2] Erbe M, Hayes BJ, Matukumalli LK, Goswami S, Bowman PJ, Reich CM, et al. Improving accuracy of genomic predictions within and between dairy cattle breeds with imputed high-density single nucleotide polymorphism panels. Journal of Dairy Science. 2012 Jul;95(7):4114–4129. Available from: https://linkinghub.elsevier.com/retrieve/pii/S0022030212003918.

[3] MacLeod IM, Bowman PJ, Vander Jagt CJ, Haile-Mariam M, Kemper KE, Chamberlain AJ, et al. Exploiting biological priors and sequence variants enhances QTL discovery and genomic prediction of complex traits. BMC Genomics. 2016 Dec;17(1):144. Available from: http://www.biomedcentral.com/1471-2164/17/144.