BayesR Program Documentation

Gerhard Moser
September 14, 2016

Overview........3

Models........3

Algorithms........3

Download........5

Compiling and running the programs........5

Options........6

Data files........7

Genotypes........7

Phenotypes........7

Covariates........8

A priori information for variance components........8

Dirichlet prior........8

Mixture model........8

Group specific priors and variance components........9

MCMC sampling........10

Prediction........10

Reduced MCMC update........11

Detailed SNP information........11

Random seed........11

SNP block options........11

Output Files........12

Log file........12

Predicted phenotypes........12

Covariates (optional)........12

Allele frequency........12

SNP effects........12

Model summary........13

Posterior samples........13

Additional SNP information (optional)........14

Examples........14

Example 1........14

Example 2........14

Example 3........15

References........15

Overview

The BayesR software implements a Bayesian mixture model for the analysis of complex traits using Markov chain Monte Carlo (MCMC). It simultaneously identifies associated SNPs, estimates the genetic variance explained by SNPs, describes the genetic architecture of the trait and predicts phenotype from SNP genotypes. Details on the method can be found in [1, 2]. The software supports multi-component modeling, including use of location specific priors [3], and to partition SNP heritability [4-6]. It is computationally efficient and can be applied to large GWAS data sets.

Models

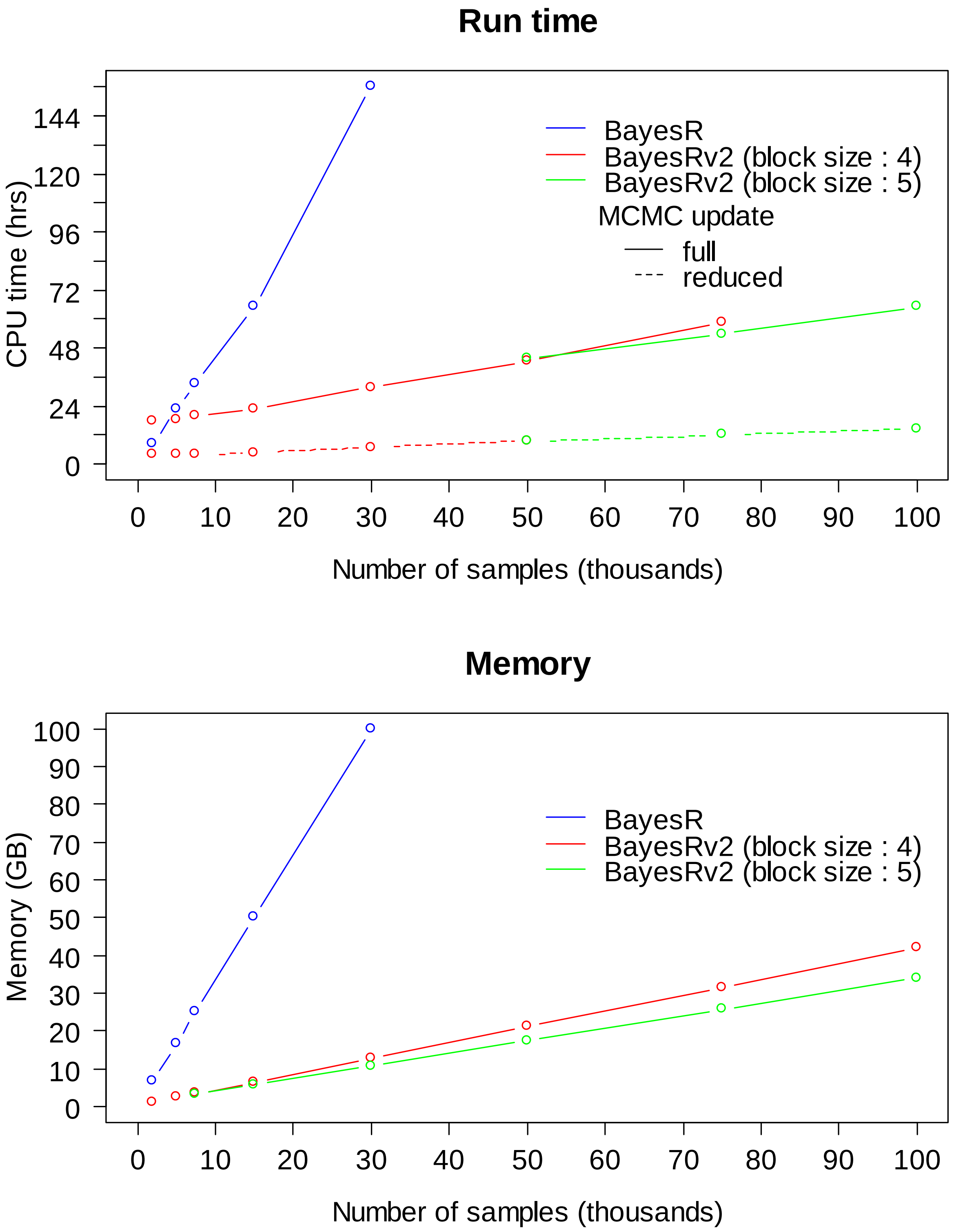
BayesR can fit univariate mixed models of the following form:

$$\mathbf{y} = \mathbf{1}_n \mu + \mathbf{W}\boldsymbol{\alpha} + \mathbf{X}\boldsymbol{\beta} + \varepsilon ,$$

where **y** is a *n*-dimensional vector of phenotypes, $\mathbf{1}_\mathrm{n}$ is a *n*-dimensional vector of ones, *μ* is the general mean, **W** is *n*×*c* matrix of fixed effects, **α** is c-dimensional vector of the corresponding coefficients, **X** is an *n*×*p* matrix of genotypes. The vector **β** is a *p*-dimensional vector of SNP effects and *ε* is a *n*-dimensional vector of residuals, $\varepsilon \sim N\left(0, \mathbf{I}\sigma_e^2\right)$ with **I** being a *n*×*n* identity matrix.

Algorithms

BayesR uses a Gibbs sampler for drawing samples from the posterior density. This approach is very flexible but computationally demanding. In the previous version of the software (available in the old/ subdirectory) the scaled and centered genotype matrix was stored in memory limiting the size of data sets that can be analysed. A more efficient implementation (bayesRv2) in terms of computing times and memory requirements can be selected at compile time and an executable is provided in the /binary subdirectory. The increased performance is largely due to a) updating effects across multiple SNPs in blocks [7] and the use of multiple CPUs, b) no longer storing the genotype matrix in memory. Further significant reduction in computing time can be achieved using a 'reduced update' strategy [2]. This option typically provides speedup of a factor of ~ 4-5.



Computational performance of various BayesR implementations
The number of SNPs was 446,907.

Download

The software is distributed under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version. You can download the software from https://github.com/syntheke/bayesR.

Compiling and running the programs

The software consists of three FORTRAN95 source code files.

BayesR: main program

RandomDistributions.f90: module containing various random number generators

baymods.f90: support module for BayesR containing common variables and routines

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

```
gfortran -o bayesR -O2 -cpp RandomDistributions.f90 baymods.f90 bayesR.f90
gfortran -o bayesRv2 -O2 -cpp -Dblock -fopenmp RandomDistributions.f90 \
baymods.f90 bayesR.f90
ifort -o bayesR -O3 -fpp RandomDistributions.f90 baymods.f90 bayesR.f90
ifort -o bayesRv2 -O3 -fpp -Dblock -openmp -static RandomDistributions.f90 \
baymods.f90 bayesR.f90
```

The supplied binaries were compiled with Intel Fortran 15.0. We found that using the Intel Fortran compiler generates code that runs at least twice as fast compared to gfortran.

Assuming the executable file is in the path of the user, the program can then be run using:

```
bayesR -bfile [prefix] -out [prefix]
```

where "`-bfile [prefix]`" specifies the PLINK binary ped files prefix and "`-out [prefix]`" specifies the output file prefix. This runs BayesR with its default options, but the user can change most parameters.

Options

Use the "`-help`" option to display a list of options with a short description and their default values (if any).

```
bayesR -h
argument   type       description                              default
-bfile     [prefix]   prefix PLINK binary files
-file      [prefix]   prefix flat input files
-out       [prefix]   prefix for output
-n         [num]      phenotype column                         1
-vara      [num]      SNP variance prior                       0.01
-vare      [num]      error variance prior                     0.01
-dfvara    [num]      degrees of freedom Va                    -2.0
-dfvare    [num]      degrees of freedom Ve                    -2.0
-delta     [num]      prior for Dirichlet                      1,1,1,1
-msize     [num]      number of SNPs in reduced update         0
-mrep      [num]      number of full cycles in reduced update  5000
-numit     [num]      length of MCMC chain                     50000
-burnin    [num]      burnin steps                             20000
-thin      [num]      thinning rate                            10
-ndist     [num]      number of mixture distributions          4
-gpin      [num]      effect sizes of mixtures (% x Va)        0.0,0.000 1,0.001,0.01
-seed      [num]      initial value for random number          0
-predict   [flag]     perform prediction                       f
-snpout    [flag]     output detailed SNP info                 f
-permute   [flag]     permute order of SNP                     f
-model     [filename] model summary file (for prediction)
-freq      [filename] SNP frequency file (for prediction)
-param     [filename] SNP effect file (for prediction)
-covar     [filename] design matrix for fixed effects
-alpha     [filename] Fixed effects estimates file (prediction)
-snpmodel  [filename] grouped effects SNP model
-varcomp   [filename] vara priors file (when vara >1)
-segments  [filename] segment priors file (when nseg >1)
-blocksize [num]      Number of SNP in rhs block               4
-nthreads  [num]      Number of threads                        4
```

Data files

The program requires input files for genotype, phenotype and (optionally) fixed effects. Genotype and phenotype input can be either in PLINK binary format or as flat files.

PLINK files: “–bfile [prefix]” specifies the prefix of the PLINK input files. Requires ‘*.bim’ and ‘*.fam’ files to determine the number of SNPs and the number of individuals and a ‘*.bed’ file for the genotype information.

Flat files: “–file [prefix]” defines the prefix when using the flat input files and requires three input files with extension ‘*.txt’, ‘*.phe’ and ‘*.map’ for genotypes, phenotypes and SNP information. The order of individuals in the genotype file must match the individuals in the phenotype file. The ‘*map’ file is only used to determine the number of SNPs and must contain one row per SNP.

Genotypes

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. PLINK ‘*.bed’ files must be in the default-SNP major mode. Flat genotype files ‘*.txt’ should have as many rows as there are individuals, each line containing the SNP information of single individual without a delimiter (blank) between consecutive SNPs (i.e. length of each line equals the number of SNPs). Genotypes must be encoded as 0, 1 or 2 copies of a reference allele, missing values are coded as ‘3’.

Phenotypes

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. Flat phenotype files ‘*.phe’ should provide trait information for each individual in a single line, with traits separated by blanks. The default (“–n 1”) uses the values in the 1st column. Missing phenotypes (or phenotypes to be predicted) must be coded as ‘NA’.

Covariates

Covariates can be included by providing a file with the design matrix for the fixed effects ("`-covar [filename]`"). Categorical predictors must be in form of indicator variables (0/1) for each level of the factor. The base R function *model.matrix* can be used for pre-processing. BayesR includes an intercept by default, hence the design matrix should not contain an intercept term. Flat priors are assigned to the regression coefficients for the fixed effects.

A priori information for variance components

Prior inverted-chi squared distribution can be specified for σ_g^2 and σ_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_g^2=heritability*σ_p^2.

Dirichlet prior

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 pseudo-observations for classes 1 to 3 of a 3-component mixture model, "`-delta 2`" sets the prior to 2 for all mixture components.

Mixture model

The BayesR model assumes that the true SNP effect is derived from a series of normal distributions. The default model uses 4 mixture distributions with SNP variances of 0, 0.0001, 0.001 and 0.01, so that the variance (S) of the *j*th SNP has 4 possible values: S1=0, S2=0.0001*Vg, S3=0.001*Vg, S4=0.01*Vg. 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*Vg, S3=0.05*Vg.

Group specific priors and variance components

The software provides flexibility in assigning SNPs to more than one mixing distribution and/or variance component. Specifying location specific priors and a single variance component will implement the model described in Brondum *et al.* (2012). Grouped effects models like partitioning across functional categories (*e.g.* Gusev et al., 2014) or dissecting genetic architecture by MAF (e.g. Lee et al., 2012) can also be specified . Assigning SNPs to multiple mixture and variance components requires a file (`-snpmodel [filename]`) in which each white-space delimited line contains the index (numeric) of the mixture component followed by the index of the variance component to which the SNPs belong. Example1 defines a model with location specific priors (column 1) and common variance, in example2 each group SNP has its own varinace.

```
Example1
1 1
2 1
1 1
1 1
2 1
2 1
Example2
1 1
2 2
1 1
1 1
2 2
2 2
```

Priors for the mixture components are specified in a file (`-segments [filename]`) which contains on a single line the prior information for each component (similar to `-ndist, -gpin and -delta` options of a single component). In the example below, we assume a mixture of 2 normal distribution of SNP effects with relative variance 0 x Vg and 0.025 x Vg for the SNPs allocated to group ‘1’ (first column in snpmodel file), and assign a pseudo-observation of 1 (SNP) for each mixture class. For group ‘2’ we assign the usual mixture of four normal distributions.

```
Example
2 0 0.025 1.0 1.0
4 0 0.0001 0.001 0.01 1.0 1.0 1.0 1.0
```

Use `-varcomp [filename]`) to specify the file containing prior scale and degrees of freedom (df) for the each variance component. In the example we choose a small value for the scale to ensure that the variance of each component stays positive. Since the scale is assumed to be somewhat data specific and affect the results its value requires tuning.

```
Example
1E-23 4.0
1E-23 4.0
```

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.

Prediction

The basic usage for prediction analysis is:

```
bayesR -bfile [prefix] -out [prefix] -predict -model [filename]
-freq [filename] -param [filename]
```

where the flag “`-predict`” must be specified for prediction analysis, “`-bfile [prefix]`” specifies the PLINK binary ped files prefix, “`-out [prefix]`” specifies the output file prefix, “`-model [filename]`” specifies the file containing the estimated mean of the BayesR model (i.e prefix.model), “`-param [filename]`” specifies the file containing the estimated SNP effects (i.e. prefix.param) and “`-freq [filename]`” specifies the file containing the allele frequencies (i.e. prefix.frq). To be predicted phenotypes must be encoded ‘NA’. For example specifying “`-n 2`” will predict phenotypes of individuals with ‘NA’ in column 7 of the PLINK *.fam file. The number and order of SNPs between training and validation data set must match.

Covariates can also be included in the prediction with `-covar [filename]` containing the design matrix of fixed effects for the to be predicted individuals and `-beta [filename]` the fixed effects coefficients usually estimated from a previous training set.

Reduced MCMC update

By setting "`-msize 0`" (default), all SNPs are updated within a single iteration of the MCMC chain. To reduce computation time for larger data sets, the model size can be set to a value > 0. For example, when specifying "`-msize 500 -mrep 1000`" all SNP effects are updated for the first 1,000 cycles. Thereafter, updating effect size continues until 500 SNPs with non-zero effects have been sampled within a cycle. The order of SNPs/blocks is randomly permuted in each MCMC cycle.

Detailed SNP information

By specifying "`-snpout`" additional SNP information (potentially very large, see Output files) for each MCMC iteration is written to a file named 'prefix.snp'. The default option is to not output detailed SNP information.

Random seed

This is an integer number for seeding the random number generator. The default "`-seed 0`" seeds with the system clock.

SNP block options

Additional options in bayesRv2 are:

argument	type	description	default
-blocksize	[num]	number of SNPs in block	4
-nthreads	[num]	number of threads	4

By default `-nthreads` is set equal to the number of SNPs within a block. BayesRv2 has limited hardcoded parallelisation. The optimal number of SNPs to include in a block depends on the number of individuals. Our limited experience suggests setting the number of threads equal to the block size will work best for most situations. A block size of 4 SNPs seems to be optimal for data set with less than 75,000 individuals.

Output Files

Log file

The file name prefix is as specified by "`-out [prefix]`". 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.

Predicted phenotypes

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

```
Example
 0.8452352E-01
  0.1995048E+00
 -0.1956367E+00
NA
 -0.1150557E+00
NA
…
```

Covariates (optional)

In the presence of covariates predictions from fixed effects are added onto the genomic values to provide phenotypes (fitted response) in file 'prefix.fitted'. Estimates of fixed effects are written to file 'prefix.beta'.

Allele frequency

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.

SNP effects

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:

PIP1..k: Posterior inclusion probabilities of the SNP in mixture classes 1 to k

beta: SNP effect

```
Example
           PIP1             PIP2             PIP3            PIP4             alpha
  0.9900000E+00    0.1000000E-01    0.0000000E+00    0.0000000E+00    0.5077715E-04
  0.9860000E+00    0.1400000E-01    0.0000000E+00    0.0000000E+00   -0.3132155E-04
  0.9780000E+00    0.2200000E-01    0.0000000E+00    0.0000000E+00    0.1003181E-05
  0.9820000E+00    0.1400000E-01    0.4000000E-02    0.0000000E+00    0.9343397E-04
  0.9760000E+00    0.2400000E-01    0.0000000E+00    0.0000000E+00    0.8146890E-05
  0.9780000E+00    0.2000000E-01    0.2000000E-02    0.0000000E+00    0.2624734E-04
```

Model summary

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,…,Nkk:	number of SNPs in mixture components 1 to *k*
Pk1,…,Pkk:	proportion of SNPs in mixture component 1 to *k*
Vk1,…,Vkk:	sum of squares of SNP effects in mixture component 1 to *k*

```
Example
Mean       0.4849967E-02
Nsnp       0.5185670E+04
Va         0.2000147E+00
Ve         0.7983887E+00
Nk1        0.2826683E+06
Nk2        0.4893394E+04
Nk3        0.2623860E+03
Nk4        0.2989000E+02
Pk1        0.9819629E+00
Pk2        0.1701720E-01
```

Posterior samples

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,…,Nkk:	number of SNPs in mixture components 1 to *k*
Vk1,…,Vkk:	sum of squares of SNP effects in mixture component 1 to *k*

```
Example
Replicate        Nsnp              Va              Ve        Nk1        Nk2
      5010       3415   0.3393990E+00   0.6650676E+00     284439       2724
      5020       3127   0.3829503E+00   0.6437855E+00     284727       2422
      5030       3089   0.2489850E+00   0.7413293E+00     284765       2491
      5040       3448   0.4474014E+00   0.5840653E+00     284406       2639

Nk3        Nk4             Vk1         Vk2               Vk3            Vk4
688          3   0.0000000E+00  0.9341668E-01  0.2401173E+00  0.1302938E-01
703          2   0.0000000E+00  0.9547626E-01  0.2713150E+00  0.1900638E-01
586         12   0.0000000E+00  0.6552167E-01  0.1472237E+00  0.3183493E-01
805          4   0.0000000E+00  0.1125713E+00  0.3406390E+00  0.4072086E-02
```

Additional SNP information (optional)

Provides additional information for SNPs selected within the model (one line per iteration). 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.

```
Example
3:65391:0.265405E-03 3:65442:0.741372E-02…
```

Examples

Example 1

A small example is provided using simulated data from the 14th QTL-MAS workshop (http://jay.up.poznan.pl/qtlmas2010/index.html). The analysis was performed using the command:

```
bayesR -bfile simdata -out simout -numit 10000 -burnin 5000 \
-seed 333
```

The results in file “simout.model” should look like this:

```
Mean        0.6865465E+02
Nsnp        0.8885003E+03
Va          0.4696789E+02
Ve          0.5335499E+02
Nk1         0.9142500E+04
Nk2         0.7321833E+03
Nk3         0.8288533E+02
Nk4         0.7343167E+02
Pk1         0.9111447E+00
Pk2         0.7307228E-01
Pk3         0.8387717E-02
Pk4         0.7395278E-02
Vk1         0.0000000E+00
Vk2         0.3419144E+01
Vk3         0.3780336E+01
Vk4         0.3939889E+02
```

Example 2

This example demonstrates how to implement the ‘genome position specific priors” model described in Brondum *et al.* (2014). SNPs from Example 1 were randomly allocated into two groups, with 20% of SNPs in group 2 affecting a trait with h^2=0.5. The simulated phenotypes are in position 7 in simdata2.fam.

```
bayesR -bfile simdata2 -out simout2 -numit 10000 -burnin 5000 \
-seed 333 -n 2 -snpmodel mod2 -segment seg
```

Example 3

Here we use the data from example 2 to estimate heritabilities for both group of SNPs.

```
bayesR -bfile simdata2 -out simout3 -numit 10000 -burnin 5000 \
-seed 333 -n 2 -snpmodel mod3 -segment seg -varcomp var3
```

References

1. Erbe M, Hayes BJ, Matukumalli LK, Goswami S, Bowman PJ, Reich CM, Mason BA, Goddard ME: **Improving accuracy of genomic predictions within and between dairy cattle breeds with imputed high-density single nucleotide polymorphism panels**. *Journal of dairy science* 2012, **95**(7):4114-4129

2. Moser G, Lee SH, Hayes BJ, Goddard ME, Wray NR, Visscher PM: **Simultaneous discovery, estimation and prediction analysis of complex traits using a bayesian mixture model**. *PLoS genetics* 2015, **11**(4):e1004969

3. Brondum RF, Su G, Lund MS, Bowman PJ, Goddard ME, Hayes BJ: **Genome position specific priors for genomic prediction**. *BMC genomics* 2012, **13**:543

4. Gusev A, Lee SH, Trynka G, Finucane H, Vilhjalmsson BJ, Xu H, Zang C, Ripke S, Bulik-Sullivan B, Stahl E *et al*: **Partitioning heritability of regulatory and cell-type-specific variants across 11 common diseases**. *American journal of human genetics* 2014, **95**(5):535-552

5. Loh PR, Tucker G, Bulik-Sullivan BK, Vilhjalmsson BJ, Finucane HK, Salem RM, Chasman DI, Ridker PM, Neale BM, Berger B *et al*: **Efficient Bayesian mixed-model analysis increases association power in large cohorts**. *Nature genetics* 2015, **47**(3):284-290.

6. Calus MP: **Right-hand-side updating for fast computing of genomic breeding values**. *Genetics, selection, evolution : GSE* 2014, **46**(1):24

7. Lee SH, DeCandia TR, Ripke S, Yang J, Sullivan PF, Goddard ME, Keller MC, Visscher PM, Wray NR: **Estimating the proportion of variation in susceptibility to schizophrenia captured by common SNPs**. *Nature Genetics 2012*, **44**(3):247–250