Boerner and Tier *Genet Sel Evol (2016) 48:63*
DOI 10.1186/s12711-016-0241-x

Genetics Selection Evolution

SOFTWARE Open Access

BESSiE: a software for linear model BLUP and Bayesian MCMC analysis of large-scale genomic data

Vinzent Boerner* and Bruce Tier

Abstract

Background: The advent of genomic marker data has triggered the development of various Bayesian algorithms for estimation of marker effects, but software packages implementing these algorithms are not readily available, or are limited to a single algorithm, uni-variate analysis or a limited number of factors. Moreover, script based environments like R may not be able to handle large-scale genomic data or exploit model properties which save computing time or memory (RAM).

Results: BESSiE is a software designed for best linear unbiased prediction (BLUP) and Bayesian Markov chain Monte Carlo analysis of linear mixed models allowing for continuous and/or categorical multivariate, repeated and missing observations, various random and fixed factors and large-scale genomic marker data. BESSiE covers the algorithms genomic BLUP, single nucleotide polymorphism (SNP)-BLUP, BayesA, BayesB, BayesCπ and BayesR for estimating marker effects and/or summarised genomic values. BESSiE is parameter file driven, command line operated and available for Linux environments. BESSiE executable, manual and a collection of examples can be downloaded http://turing.une.edu.au/~agbu-admin/BESSiE/.

Conclusion: BESSiE allows the user to compare several different Bayesian and BLUP algorithms for estimating marker effects from large data sets in complex models with the same software by small alterations in the parameter file. The program has no hard-coded limitations for number of factors, observations or genetic markers.

Background

In quantitative genetics, various software packages are available for the analysis of phenotypic observations with linear mixed models, which can be categorised by the algorithm used to infer dispersion and location parameters of the modelled factors: (a) restricted maximum likelihood (REML) based software, and (b) Bayesian inference based software using Markov chain Monte Carlo (MCMC) methods (e.g. Gibbs sampling). While various REML software packages specifically designed for quantitative genetics are widely used and well documented, (e.g. ASREML [1], WOMBAT [2], DMU [3], REMLF90 [4], VCE [5]), software packages that apply the MCMC methodology are less common (e.g. GIBBSF90 and THRGIBBSF90 [4], GS3 [6], BAYESR [7], MCMCglmm [8]). The relatively small number of MCMC software packages for quantitative geneticists may reflect the disadvantage of MCMC methodology in terms of processing time. In addition, large-scale genomic marker data [e.g. single nucleotide polymorphisms (SNPs)] that emerge in the late 2000 can easily be accommodated in existing REML software via approaches such as single marker regression, genomic best linear unbiased prediction (GBLUP) [9] or SNP-BLUP [10]. By contrast, several Bayesian algorithms for sampling dispersion and location parameters of genomic markers have been proposed (e.g. "BayesA", "BayesB", "BayesCπ", "BayesR" [11–13]), which differ only slightly but require adjustments in the source code, thus making it more difficult to develop and maintain a software package which covers all algorithms.

The aim of this article is to describe the software BESSiE which is designed for uni- and multivariate BLUP and

*Correspondence: vboerner@une.edu.au
Animal Genetics and Breeding Unit, University of New England, Armidale 2351, Australia

BioMed Central

© 2016 The Author(s). This article is distributed under the terms of the Creative Commons Attribution 4.0 International License (http://creativecommons.org/licenses/by/4.0/), which permits unrestricted use, distribution, and reproduction in any medium, provided you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made. The Creative Commons Public Domain Dedication waiver (http://creativecommons.org/publicdomain/zero/1.0/) applies to the data made available in this article, unless otherwise stated.

Bayesian analysis of linear mixed models in quantitative genetics allowing for various factors, algorithms, large-scale genomic data and both continuous as well as categorical observations.

Software description

Program modes

BESSiE allows for two different modes, “BLUP” and “GIBBS”.

Mode “BLUP” performs a best linear unbiased analysis of the specified linear mixed models given the observed data and supplied variances of random factors. The results are best linear unbiased estimations for levels of fixed factors and best linear unbiased predictions for levels of random factors.

Mode “GIBBS” performs a Bayesian analysis of the specified linear mixed model given the observed data using supplied variances of random factors as starting values or as prior knowledge in an MCMC Gibbs sampling approach, which is expanded by Metropolis–Hasting steps if required. The results for factor levels and for co-variances are draws from their conditional posterior distributions as given in [14]. In addition, for factors that model genetic markers, the results are draws from a posterior distribution as specified in [11] (“BayesA” and “BayesB”), in [12] (“BayesCπ) or in [13] (“BayesR”).

Models

The super-set model to be fitted in BESSiE may be written as:

$$\begin{pmatrix} \mathbf{y}_1 \\ . \\ \mathbf{y}_n \end{pmatrix} = \begin{pmatrix} \mathbf{X}_1 & . & 0 \\ . & . & . \\ 0 & . & \mathbf{X}_n \end{pmatrix} \begin{pmatrix} \mathbf{b}_1 \\ . \\ \mathbf{b}_n \end{pmatrix} + \begin{pmatrix} \mathbf{Z}_{1,1} & . & \mathbf{Z}_{1,k} & . & 0 & 0 & 0 \\ . & . & . & . & . & . & . \\ 0 & 0 & 0 & . & \mathbf{Z}_{n,1} & . & \mathbf{Z}_{n,k} \end{pmatrix} \begin{pmatrix} \mathbf{u}_{1,1} \\ . \\ \mathbf{u}_{n,k} \end{pmatrix} + \begin{pmatrix} \mathbf{Q}_1\mathbf{M} & . & 0 \\ . & . & . \\ 0 & . & \mathbf{Q}_n\mathbf{M} \end{pmatrix} \begin{pmatrix} \mathbf{g}_1 \\ . \\ \mathbf{g}_n \end{pmatrix} + \begin{pmatrix} \mathbf{e}_1 \\ . \\ \mathbf{e}_n \end{pmatrix}$$

where $(\mathbf{y}_1, ., \mathbf{y}_n)'$, $(\mathbf{b}_1, ., \mathbf{b}_n)'$, $(\mathbf{u}_{1,1}, ., \mathbf{u}_{n,k})'$, $(\mathbf{g}_1, ., \mathbf{g}_n)'$ and $(\mathbf{e}_1, ., \mathbf{e}_n)'$ are vectors of phenotypic observations of linear or categorical scale (including repeated and/or missing observations), fixed effects, random non-marker effects [1..k] and random marker effects, $\mathbf{X}$, $\mathbf{Z}$ and $\mathbf{Q}$ are matrices relating the effects to their respective observations, $\mathbf{M}$ is a matrix of marker genotypes of dimension “number of genotyped individuals” × “number of markers” and the subscripts are for trait 1 to n. Values in $\mathbf{X}$ may be dummy variables or linear co-variables, where for the latter the order of polynomial regression is user-defined. Values in $(\mathbf{u}_1, ., \mathbf{u}_n)$ are assumed to be distributed $N([0, ., 0]', \mathbf{A} \otimes \Sigma)$, $N([0, ., 0]', \mathbf{G} \otimes \Sigma)$, $N([0, ., 0]', \mathbf{I} \otimes \Sigma)$ or $N([0, ., 0]', \mathbf{K} \otimes \Sigma)$, where $\mathbf{A}$ is the pedigree-derived numerator relationship matrix, $\mathbf{G}$ is a relationship matrix derived from genetic markers, $\mathbf{I}$ is an identity matrix, $\mathbf{K}$ is an unknown, but symmetric and positive definite matrix of dimension “number of factor levels” × “number of factor levels” provided by the user, and Σ is a co-variance matrix of factors. Note that all random non-marker effects can be fitted together.

Effects of genetic markers $(\mathbf{g}_1, ., \mathbf{g}_n)'$ can be obtained from “BayesA” and “BayesB” [11], “BayesCπ ” [12], “BayesR” [13] or ridge regression SNP BLUP [10]. For “BayesA”, “BayesB” and “BayesCπ”, default values for parameters of the algorithms and prior distributions of marker variances are derived from the related publications, but can also be specified by the user.

If the co-variance structure of a factor is $\mathbf{G} \otimes \Sigma$, where $\mathbf{G}$ is a genomic relationship matrix, $\mathbf{G}$ can be pre-calculated by the user and read from a file, or can be calculated by the program from a file of genomic markers. In the latter case, two methods are provided to calculate $\mathbf{G}$ described in [9] and [15], the latter being similar to the second method in [9] except for the diagonal elements.

Residuals are assumed to be distributed $N([0, ., 0]', \mathbf{I} \otimes \mathbf{R})$, where $\mathbf{R}$ is the residual co-variance matrix of dimension n × n. However, to account for observations with different residual variances (e.g. de-regressed breeding values), a co-variance $\boldsymbol{\Omega}$ can be modelled, where $\boldsymbol{\Omega}$ is a matrix of diagonal blocks containing $\boldsymbol{\omega}_1\sigma^2_{e_1}$ to $\boldsymbol{\omega}_n\sigma^2_{e_n}$ in the diagonal elements of the diagonal blocks, and $\sqrt{\boldsymbol{\omega}_1\boldsymbol{\omega}_n}\sigma_{e_{1,n}}$ in the diagonal elements of the off-diagonal block which links trait 1 and trait n, where $\boldsymbol{\omega}_1$ and $\boldsymbol{\omega}_n$ are vectors of weights for trait 1 and n, and $\sigma^2_{e_1}$, $\sigma^2_{e_n}$ and $\sigma_{e_{1,n}}$ are the residual variances and co-variance of the traits.

In multivariate analysis using “BayesA”, “BayesB”, “BayesCπ ” or “BayesR” effects of genetic markers are estimated from

$$\left(\begin{bmatrix} \mathbf{Q}_1\mathbf{M} & . & 0 \\ . & . & . \\ 0 & . & \mathbf{Q}_n\mathbf{M} \end{bmatrix}' \mathbf{R}^{-1} \begin{bmatrix} \mathbf{Q}_1\mathbf{M} & . & 0 \\ . & . & . \\ 0 & . & \mathbf{Q}_n\mathbf{M} \end{bmatrix} + \begin{bmatrix} \boldsymbol{\Sigma}_1 & . & 0 \\ . & . & . \\ 0 & . & \boldsymbol{\Sigma}_n \end{bmatrix}^{-1} \right) \begin{pmatrix} \mathbf{g}_1 \\ . \\ \mathbf{g}_n \end{pmatrix} = \begin{pmatrix} \mathbf{Q}_1\mathbf{M} & . & 0 \\ . & . & . \\ 0 & . & \mathbf{Q}_n\mathbf{M} \end{pmatrix}' \mathbf{R}^{-1} \left(\begin{bmatrix} \mathbf{y}_1 \\ . \\ \mathbf{y}_n \end{bmatrix} - \begin{bmatrix} \mathbf{X}_1 & . & 0 \\ . & . & . \\ 0 & . & \mathbf{X}_n \end{bmatrix} \begin{bmatrix} \mathbf{b}_1 \\ . \\ \mathbf{b}_n \end{bmatrix} - \begin{bmatrix} \mathbf{Z}_{1,1} & . & 0 \\ . & . & . \\ 0 & . & \mathbf{Z}_{n,k} \end{bmatrix} \begin{bmatrix} \mathbf{u}_{1,1} \\ . \\ \mathbf{u}_{n,k} \end{bmatrix} \right)$$

where $\mathbf{\Sigma}_1$ to $\mathbf{\Sigma}_\mathrm{n}$ are diagonal matrices of dimension "number of markers" × "number of markers" of which elements contain the marker variances generated according to the Bayesian method specified for trait 1 to n. Thus, the co-variances between the effects of a genetic marker on trait 1 to n are assumed to be zero. The user should be aware that modelling markers in a multivariate analysis that assumes no co-variance between effects of a single marker on different phenotypes may lead to spurious results if a polygenic (pedigree-based) component is not included in the model.

Program input

BESSiE reads all necessary information from a parameter file, which also contains names and locations of other input files. These might be files containing data, genotypes, pedigree, co-variances and other matrices. Program information in the parameter file is identified by case sensitive keywords, and the block structure within these keywords are nested, which ensures robustness against input errors, easy extension for new input parameters, and allows for extensive commenting outside the block structure (see Fig. 1).

BESSiE data files must contain only numeric input and assumes the user will be able to renumber all modelled factors. However, BESSiE conducts an extensive data check and will provide informative error messages in case it detects any inconsistency in factor numbering.

If BESSiE is used to estimate effects of genetic markers via one of the six above mentioned implemented algorithms, it expects that markers are coded "0", "2" and "1" for both the homozygous and the heterozygous genotype. Thus, missing markers within genotypes are not accommodated for. Markers are provided via an ASCII file containing one line per genotype and as many columns as markers. Thus, genotypes of different markers must not be separated by any character, which allows for a reduced file sizes. However, BESSiE checks for valid marker coding while reading through that file.

Program output

BESSiE generates various outputs in ASCII or Fortran binary files where the desired output can be chosen by the user. In mode "BLUP", the output may include solutions for factor levels conditional on the variances provided. In mode "GIBBS", the output may include draws from conditional posterior distributions of factor levels and factor co-variances. For random marker effects modelled by "BayesA", "BayesB", "BayesCπ" or "BayesR", effects and variances for each marker as well as the total variance explained by genetic markers and summarised genomic values for each genotype can optionally be written in the file. Moreover, for algorithms "BayesCπ" and "BayesR", the output may include the draws from Beta/Dirichlet conditional posterior distributions assigning probabilities to the distributions from which markers may come from, and marker-to-distribution assignment statistics. By default BESSiE also generates an extensive log file which contains informative messages in case of input errors and information about the current state of the program while a Bayesian analysis is running, for example.

Technical details

For Bayesian analysis, BESSiE uses a blocked Gibbs sampler as described in [14], but only for fixed and random non-marker effects. That is, random non-marker effects are sampled from their assigned distributions. The right-hand side of the mixed model equation (MME) is corrected for these draws and the MME is subsequently solved. Draws are added back to the related MME solutions and the resulting values can be regarded as draws from the conditional posterior of each random non-marker effect. Effect sizes and variances of genetic markers modelled by "BayesA", "BayesB", "BayesCπ" or "BayesR" are obtained from a single site Gibbs sampler using phenotypic observations corrected for fixed and random non-marker effects. For solving the MME in a BLUP analysis and in blocked Gibbs sampling, BESSiE uses a preconditioned gradient solver. This may slow down the number of Gibbs sampler cycles per second if the data set is very large, but it is assumed to accelerate convergence drastically [16, 17]. BESSiE does not set up the coefficient matrix of the MME. While this prevents exploitation of parallel processing, computer memory requirements are kept to a manageable level even for genome-wide association studies including whole-genome sequences and "BLUP" estimations including millions of animals. However, parts of the preconditioned gradient solver steps are parallelised. The algorithm that is used to obtain dispersion and location parameters when trait observations are of categorical scale is described in [18] and [19].

Speed, memory requirements, implementation and availability

BESSiE has no hard coded limitations in terms of number of traits, factors, genotypes and markers, and has been tested on very large data sets.

As an example, a bi-variate analysis with 4420 individuals genotyped for 510,174 SNPs, 19,549 individuals in the pedigree, seven fixed effects and a polygenic random effect per trait, and SNP effects modelled according to "BayesR" with four distributions required ~4.3 GB of RAM and about seven real time seconds on an Intel(R) Core(TM) i7-3770 processor to sample all location and dispersion parameters once.

```
space for ‘‘un-commented’’comments .................
.....................................................
BEGIN PARAMETERFILE !start to read here
  !this is a comment within PARAMETERFILE
  MODE: GIBBS !we want Gibbs sampling
  INPUT: PEDIGREE,TRAITS,GENOTYPES !check for PEDIGREE, TRAITS and GENOTYPES
  BEGIN PEDIGREE !start definition of PEDIGREE
     FILE: Pedigree.txt !read pedigree from here
  END PEDIGREE !end definition of PEDIGREE
  BEGIN GENOTYPES !start definition of GENOTYPES
    FILE: MyGenotypes.txt !read genotypes from here
  END GENOTYPES !end definition of GENOTYPES
  BEGIN TRAITS !start definition TRAITS
    FILE: Data.txt !read data from here
    NAMES: CalvingEase !define trait names
    CAT: CalvingEase !this trait has categorical phenotypes
    BEGIN CalvingEase !start definition of trait named ‘‘calving ease’’
      OBSERVPOS: 1 !define the observation column in the data file
      RESVARPOS: 2 !define the position of the residual variance in the co-variance matrix
      MODEL: mean+weight+animal+snp !define the model/factors
      BEGIN mean !start definition of factor named ‘‘mean’’
        TYPE: FIX !this is a fixed dummy variable
        FILEPOS: 2 !define the factor column in the data file
      END mean
      BEGIN weight !start definition of factor named ‘‘weight’’
        TYPE: FIX !this is a fixed variable
        FILEPOS: 2 !define the factor column in the data file
        COV: T !switch from dummy variable to co-variable
        EXP: 3 !set the exponent of the polynomial
      END weight
      BEGIN animal !start definition of factor named ‘‘animal’’
        TYPE: RAN !this is a random variable
        STRUCTURE: NRM !this factor is related to the pedigree
        FILEPOS: 3 !define the factor column in the data file
        COVARPOS: 2 !define the position of the factor variance in the co-variance matrix
        PRINT: EA !get me the sampled factor level effects in ASCII
      END animal
      BEGIN snp !start definition of factor named ‘‘snp’’
        TYPE: RAN !this is a random variable
        STRUCTURE: SNP !this factor is related to the genotypes
        METHOD: BAYESR !this factor should be modelled according ‘‘BayesR’’
        TOTALSNPVAR: 25 !maximum variance possibly explained by the marker
        DISTFILE: BayesR.txt !read the parameter the ‘‘BayesR’’ distributions from here
        FILEPOS: 4 !define the factor column in the data file
      END snp
    END CalvingEase
    BEGIN COVAR
      FILE: Covar.txt !read the co-variance matrix from here
      PRINT: VA !get me the sampled variances in ASCII
    END COVAR
  END TRAITS
  BEGIN GIBBSSAMPLER !define the sampler here
    CYCLES: 10000 !number of cycles
    BURNIN: 0
    PRINTINT: 1 ! printing frequence for samples
  END GIBBSSAMPLER
END PARAMETERFILE
more space for ‘‘un-commented’’ comments ...................
.............................................................
```

Fig. 1 Example parameter file

Another example is a uni-variate analysis of publicly available mouse data (http://mus.well.ox.ac.uk/mouse/) consisting of 1940 phenotypes and genotypes, where genotypes contained 8516 SNP genotypes, and a model including only the mean and a random marker factor modelled by the "BayesR" algorithm with four distributions. When executed on a notebook with an Intel(R) Core(TM) i7-2637M processor, BESSiE used ~72 MB of RAM and need 5 real time milliseconds to sample all location and dispersion parameters once.

BESSiE is written in Fortran 2008, command line operated, parameter file driven and comes with an extensive manual. It is available for 64bit Unix-like operation systems only. BESSiE uses the Intel Math Kernel library [20] for random number generation and matrix operations and is therefore optimised for Intel architecture. However, it will run on AMD architecture but run time may increase.

BESSiE comes free of charge for the scientific community, but users are required to credit its use in any publication. Commercial users must contact the authors. BESSiE executable, manual and a collection of examples can be downloaded from http://turing.une.edu.au/~agbu-admin/BESSiE/. BESSiE is under ongoing development, and due to the number of features, some combinations of algorithms and/or modelled factors may not have been tested thoroughly. Thus, users use BESSiE at their own risk.

Authors' contributions

VB designed and wrote the software and the manuscript. BT revised the manuscript. Both authors read and approved the final manuscript.

Acknowledgements

This work was funded by Meat and Livestock Australia (Project B.BFG.0050).

Competing interests

The authors declare that they have no competing interests.

Received: 17 February 2016 Accepted: 25 August 2016
Published online: 02 September 2016

References

1. Gilmour AR, Gogel BJ, Cullis BR, Thompson R. ASReml user guide release 3.0; 2009.
2. Meyer K. WOMBAT: a tool for mixed model analyses in quantitative genetics by restricted maximum likelihood (REML). J Zhejiang Univ Sci B. 2007;8(11):815–21. doi:10.1631/jzus.2007.B0815.
3. Madsen P, Jensen J, Labouriau R, Christensen OF, Sahana G. DMU-a package for analyzing multivariate mixed models in quantitative genetics and genomics. In: Proceedings of the 10th world congress of genetics applied to livestock production. 18–22 August 2014; Vancouver; 2014. https://asas.org/docs/default-source/wcgalp-posters/699_paper_9580_manuscript_758_0.pdf?sfvrsn=2.
4. Misztal I, Tsuruta S, Strabel T, Auvray B, Druet T, Lee DH. BLUPF90 and related programs (BGF90). In: Proceedings of the 7th world congress of genetics applied to livestock production. 19–23 August 2002; Montpellier; 2002.
5. Groeneveld E, Kovac M, Mielenz N. VCE user's guide and reference manual version 6.0; 2010.
6. Legarra A, Ricard A, Filangi O. GS3: Genomic selection, gibbs sampling, Gauss-Seidel (and BayesCpi).; 2011. http://genoweb.toulouse.inra.fr/%7Ealegarra/.
7. 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 Genet. 2015;11(4):e1004969. doi:10.1371/journal.pgen.1004969.
8. Hadfield JD. MCMC methods for multi-response generalized linear mixed models: the MCMCglmm R package. J Stat Softw. 2010;33(2):1.
9. VanRaden PM. Efficient methods to compute genomic predictions. J Dairy Sci. 2008;91(11):4414–23. doi:10.3168/jds.2007-0980.
10. Piepho HP. Ridge regression and extensions for genomewide selection in maize. Crop Sci. 2009;49(4):1165–76.
11. Meuwissen TH, Hayes BJ, Goddard ME. Prediction of total genetic value using genome-wide dense marker maps. Genetics. 2001;157(4):1819–29.
12. Habier D, Fernando RL, Kizilkaya K, Garrick DJ. Extension of the Bayesian alphabet for genomic selection. BMC Bioinformatics. 2011;12:186. doi:10.1186/1471-2105-12-186.
13. 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. J Dairy Sci. 2012;95(7):4114–29. doi:10.3168/jds.2011-5019.
14. Sorensen D, Gianola D. Likelihood, Bayesian, and MCMC methods in quantitative genetics. In: Dietz K, Gail M, Krickeberg K, Samet J, Tsiatis A, editors. New York: Springer; 2002.
15. Yang J, Benyamin B, McEvoy BP, Gordon S, Henders AK, Nyholt DR, et al. Common SNPs explain a large proportion of the heritability for human height. Nat Genet. 2010;42(7):565–9. doi:10.1038/ng.608.
16. Liu JS. The collapsed Gibbs sampler in Bayesian computations with applications to a gene regulation problem. J Am Stat Assoc. 1994;89(427):958–66.
17. Liu JS, Wong WH, Kong A. Covariance structure of the Gibbs sampler with applications to the comparisons of estimators and augmentation schemes. Biometrika. 1994;81(1):27–40.
18. Sorensen DA, Andersen S, Gianola D, Korsgaard I. Bayesian inference in threshold models using Gibbs sampling. Genet Sel Evol. 1995;27:229–49.
19. Albert JH, Chib S. Bayesian analysis of binary and polychotomous response data. J Am Stat Assoc. 1993;88:669–79.
20. Intel; 2014. https://software.intel.com/en-us/articles/intel-math-kernel-library-documentation.

Submit your next manuscript to BioMed Central and we will help you at every step:

- We accept pre-submission inquiries
- Our selector tool helps you to find the most relevant journal
- We provide round the clock customer support
- Convenient online submission
- Thorough peer review
- Inclusion in PubMed and all major indexing services
- Maximum visibility for your research

Submit your manuscript at
www.biomedcentral.com/submit

BioMed Central