**JWAS: Julia implementation of Whole-genome Analysis Software**

Hao Cheng\(^1\), Rohan Fernando\(^2\) & Dorian Garrick\(^3\)

\(^1\)Department of Animal Science, University of California, Davis, US
\(gtlcheng@ucdavis.com\) (Corresponding Author)
\(^2\)Department of Animal Science, Iowa State University, Ames, Iowa, US
\(^3\)School of Agriculture, Massey University, New Zealand

**Summary**

Genome-wide high-density molecular markers (e.g., SNPs) are increasingly being incorporated into animal and plant breeding programs to speed up genetic improvement through genomic prediction. The same statistical models can also be used for genome-wide association studies. Bayesian multiple-regression methods are widely used in genomic prediction with complete genomic data (all phenotyped individuals in the analysis are genotyped). These methods have been extended to accommodate incomplete genomic data (some phenotyped animals not genotyped), simultaneously using all available pedigree, phenotypic and genomic information (“single-step” Bayesian methods). We have developed a well-documented software tool called JWAS (acronym for "Julia Whole-genome Analysis Software") in a relatively new scientific programming language, Julia, which approaches the computing speed of compiled languages such as C, C++ or Fortran, but has the benefits of dynamic languages such as R or Python.

JWAS is a well-documented software platform based on Julia and an interactive Jupyter notebook for analyses of general univariate and multivariate Bayesian mixed effects models. These models are especially useful for, but not limited to, routine single-trait and multi-trait genomic prediction and genome-wide association studies using either complete or incomplete genomic data ("single-step" methods). Currently, JWAS provides broad scope of analyses, e.g., a wide collection of Bayesian methods for whole-genome analyses, including shrinkage estimation and variable selection methods. We believe the friendly user interface and fast computing speed of JWAS will provide power and convenience for users in both industry and academia to analyse large datasets. Further, as a well-documented open-source software tool, JWAS will also be used by a group of active community members, who will contribute to the source code and help maintain the project. Junior scientists can understand and learn the methodologies for whole-genome analyses by using JWAS and reading the tutorials and source code. JWAS is available with source code and documentation at http://QTL.rocks.

**Keywords:** Bayesian mixed effects model, Julia, multivariate, complete genomic data, incomplete genomic data

**Introduction**

The discovery of genome-wide high-density molecular markers (e.g., single-nucleotide polymorphisms, SNPs) and effective ways to assay them has revolutionized genetic analyses of quantitative traits in animal (e.g., VanRaden, 2008) and plant breeding (e.g., Crossa et al.,
Selection based on genomic prediction incorporating marker effects from whole-genome data was proposed to by Meuwissen et al. (2001) for animal and plant breeding, and has become an important application of whole-genome data in agriculture. In genomic prediction, all the marker effects are estimated simultaneously, and used to predict the breeding values. Whole-genome data is also used in genome-wide association studies (GWAS) to detect quantitative trait locus (QTL) regions. In GWAS, the association between molecular markers and phenotypes is assessed, where each marker (e.g., Manolio et al., 2009) or genomic window (e.g., Hayes et al., 2010) is tested one at a time (Manolio et al., 2009) or simultaneously (Hayes et al., 2010).

Genomic analyses with all individuals genotyped (hereafter referred to as complete genomic data) are relatively straightforward, but in general, the number of individuals with genomic information is only a subset of the individuals represented in the population with pedigree and phenotypic information (hereafter referred to as incomplete genomic data). "Single-step" methods were developed to simultaneously take advantage of all available pedigree, phenotypic and genomic information, the latter being observed on a subset of the pedigree individuals as in Legarra at al. (2009) and Fernando et al. (2014). In what is known as "single-step" genomic BLUP (SSGBLUP) (Legarra et al., 2009), an elegant strategy is used to construct a relationship matrix that combines genotypic and pedigree information. An alternative marker effects model was proposed by Fernando et al. (2014)) for “single-step” analyses. "Single-step" methods were shown to yield similar or higher accuracy for genotyped individuals than could be obtained by ignoring information from non-genotyped relatives (Misztal et al., 2013).

Bayesian multiple-regression methods are widely used in genomic prediction (Meuwissen et al., 2001), and can also be adapted for GWAS (Fernando et al., 2017). A wide class of Bayesian regression methods were initially proposed for genomic prediction with complete genomic data, including BayesA and BayesB (Meuwissen et al., 2001), BayesC (Kizilkaya et al., 2010), BayesCpi (Habier et al., 2011), Bayesian Lasso (Gianola, 2013) and BayesR (Erbe et al., 2012). These methods only differ in the priors used for SNP effects. All these priors can be incorporated into analyses with incomplete genomic data through a class of "single-step" Bayesian regression (SSBR) methods first introduced by Fernando et al., (2014). SSBR methods yield higher prediction accuracies than SSGBLUP with some real datasets (Lee et al., 2017) and provide computational benefits when many animals are genotyped (Fernando et al., 2016a; 2016b).

Statistical inference in these Bayesian methods are often based on Markov chain Monte Carlo (MCMC) techniques, which can be computationally intensive for large-scale applications. Such analyses can be implemented with most dynamic languages such as R or Python with very readable code that is easy to understand, modify and extend. However, dynamic languages are often too slow for analyses requiring MCMC algorithms for large data sets. Compiled languages such as C, C++, or Fortran are usually used to implement Bayesian analyses because of computational efficiencies. Applications developed with compiled languages, however, are usually hard to understand, modify, extend and maintain, and can be difficult to deploy across different operating systems. Thus, a single programming language designed to address the need of code readability and high-performance computing is desired. Julia (Bezanson et al., 2012), a relatively new scientific programming language, approaches the computing speed of compiled languages but has the benefits of dynamic languages.

Recently, we have developed a software package in Julia for genomic analyses including genomic prediction and GWAS, called JWAS (acronym for “Julia Whole-genome
Analyses Software”). The main objective of this paper is to introduce a well-documented software platform based on Julia and interactive Jupyter notebooks for fitting general univariate and multivariate Bayesian mixed effects models, especially useful for, but not limited to, single-trait and multi-trait genomic prediction and genome-wide association studies using either complete or incomplete genomic data (i.e. "single-step" methods).

**Methods**

JWAS allows a wide collection of Bayesian methods for whole-genome analyses, including shrinkage estimation and variable selection methods, using complete and incomplete genomic data. JWAS is a more powerful, user-friendly, and open-source extension to GenSel (Fernando & Garrick, 2009), a widely-used tool for whole-genome analyses. It provides broader scope of analyses but still approaches the computing speed of GenSel (Figure 1). Comparison between JWAS and GenSel in modeling is shown in Table 1.

<table>
<thead>
<tr>
<th>Software Tools</th>
<th>JWAS</th>
<th>GenSel</th>
</tr>
</thead>
<tbody>
<tr>
<td>No limitations on fixed effects (e.g. herd-year, age, sex)</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Random effects other than markers (e.g. litter, pen)</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Random effects using pedigree information</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Random permanent environmental effects</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Use of genomic information</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Complete genomic data(^1)</td>
<td>single-trait analyses</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>multi-trait analyses</td>
<td>Yes</td>
</tr>
<tr>
<td>Incomplete genomic data(^2)</td>
<td>single-trait analyses</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>multi-trait analyses</td>
<td>Yes</td>
</tr>
<tr>
<td>Correlated residuals</td>
<td>Yes</td>
<td>No</td>
</tr>
</tbody>
</table>

\(^1\)“Complete genomic data” indicates that genotypes are available on all individuals.

\(^2\)“Incomplete genomic data” indicates that genotypes are not available on all individuals ("single-step" analyses).

There are several other major advantages to JWAS being written in Julia, over GenSel written in C++, or similar software tools written in compiled languages: 1) Julia programs can approach the computing speed of compiled languages while maintaining the benefits of dynamic languages such as R and Python. Most compute-intensive packages for genomic prediction are entirely or in part written in a compiled language such as C, C++, or Fortran, which make them harder to understand, modify, extend and maintain; 2) Julia (and therefore JWAS) allows easy use of multi-core CPU or GPU computing capabilities; 3) As demonstrated in Figure 2, JWAS has a friendly user interface compared to most existing packages for genomic prediction; 4) An interactive Jupyter notebook interface is available for JWAS, which makes it ideal for “reproducible research”. The Jupyter notebook is an open-source web application for creating and sharing documents that contain live code, equations, visualizations and explanatory text (Perez et al., 2007); 5) JWAS is completely open-source. Researchers can contribute to the project code by submitting their code through GitHub. In contrast, most packages for genomic prediction, which are written in compiled languages, are not open-source.
Example

An example to demonstrate the user interface of JWAS is in Figure 2. The interactive Jupyter notebook containing live code and explanatory text is used to demonstrate a two-trait Bayesian regression analysis fitting fixed effects (sex, age), random effects (litter), direct genetic effects (animal), maternal genetic effects (dam) and genomic information.

In figure 2, in cell 3, the data file is read on line 1, and the pedigree information is read on line 2. On line 3, the first several rows of data are shown. In cell 4, the non-genomic part of the model equation for a 2-trait analysis is defined. The effects fitted in the model for trait 1 are the intercept, sex, direct genetic effects and maternal genetic effects. The effects fitted in the model for trait 2 are the intercept, sex, age, the interaction between sex and age and direct genetic effects. In cell 5, the model is built given the model equation in cell 4. By default, all effects are treated as fixed and classed as factors (categorical variables) rather than covariates (quantitative variables). On line 2, the effect age is defined to be a covariate rather than class effect. On line 3, the litter class effect is defined as random. On line 4, direct genetic effects and maternal genetic effects are fitted as “animal” and “dam” using the inverse of the numerator relationship matrix defined from pedigree. In cell 6, the genomic part of the model is defined with the genotype file. In cell 7, a multi-trait BayesC analysis is performed with “model” and “data” as had been defined in cells 1 to 5.

Conclusion

We have developed a single-language software platform ideal for routine data analyses and "reproducible research" fitting single-trait or multi-trait genomic prediction models, as well as for GWAS, using complete or incomplete genomic data that makes it easy for our community of researchers to participate, document, maintain and extend. We believe the friendly user interface and fast computing speed of JWAS will provide power and convenience for users in industry and academia to analyse large datasets. Further, as a well-documented open-source software tool, JWAS may be used by a group of active community members, who will contribute to the source code and help maintain the project. It will also appeal to junior scientists who can understand and learn the methodologies for whole-genome analyses by using JWAS and by reading the tutorials and source code.
Figure 1. Comparison of computing time for BayesC with 50,000 iterations of MCMC using JWAS, GenSel or an R implementation of the same algorithm.
Figure 2. User interface of JWAS.
List of References


