FW: An R package for Genomic/Pedigree and Spatial analysis using the Finlay-Wilkinson Regression

Lian Lian and Gustavo de los Campos*

Abstract

The Finlay-Wilkinson Regression is a popular method among plant breeders to describe genotype by environment interaction. The standard implementation is a two-step procedure that uses environment (sample) means as covariates in a within-line ordinary least squares (OLS) regression. This procedure can be suboptimal for at least four reasons: (i) in the first step environmental means are computed without considering genotype effects, (ii) in the second step, uncertainty about the environmental means is ignored, (iii) estimation is performed regarding lines and environment as fixed effects and (iv) the procedure does not incorporate genetic (either pedigree-derived or marker derived) relationships. Su et al. proposed to address these problems using a Bayesian method that allows simultaneous estimation of environmental and genotype parameters, and allows incorporation of pedigree information. In this article we: (i) extend the model presented by Su et al. to allow integration of genomic (e.g., SNP) and spatial information, (ii) present an R package (FW) that implements these methods, and (iii) illustrate the use of the package using examples based on real data. The FW R-package implements both the standard two-step OLS method and a full Bayesian approach for Finlay-Wilkinson regression with a very simple interface. For computational efficiency, the algorithm was implemented in compiled C code. In this article we describe the methods implemented in FW and presents examples of the uses of the package.

Introduction

Plant breeders use the Finlay-Wilkinson Regression (Finlay and Wilkinson, 1963) to assess the stability of varieties across different environments. The standard implementation is a two-step procedure whereas in the first step environmental sample means are computed and in the second step intercepts and slopes of each line are estimated by regressing, within line, the performance of each line on the estimated environmental means.

The standard two step procedure has at least four potential limitations: (i) in the first step environmental means are computed without considering genotype effects, (ii) in the second step, uncertainty about the environmental means is ignored (iii) the environmental means and the variety intercepts and slopes are regarded as fixed effects (this can lead to large sampling variance), and (iv) the procedure does not offer a clear way of incorporating pedigree or molecular marker information when estimating the intercepts and slopes of the lines. These drawbacks can induce biases (especially in incomplete designs where a few lines are evaluated in each environments) and lead to large sampling variance of estimates.

A Bayesian method was proposed by Su et al. (2006) to address the limitations of the standard two-step procedure. The methodology described by Su et al.: (1) uses a Gibbs sampler that allows estimating environmental and genotype parameters jointly, (2) fully accounts for confounding and uncertainty about environmental means, (3) treats environmental means and the intercepts and slopes of the lines as random—this treatment usually perform better than ordinary least squares in terms of mean-squared error and of prediction accuracy, especially when the number of parameters to be estimated is large

relative to sample size (Copas, 1983; Frank and Friedman, 1993), and (iv) allows incorporating pedigree information into the model. Su et al. (2006) reported better performance of the Bayesian method in estimating the true parameters. In this article we extend the model proposed by Su et al. (2006) in ways that allow incorporating genomic (e.g., SNP) and spatial information.

To the best of our knowledge the methodology described by Su et al. for animal breeding applications has not been considered in plant breeding applications, and there is no publicly available user-friendly software for implementing a Bayesian FW regression. Therefore, in this article we introduce an R-package (R Development Core Team, 2011) that implements the Finlay Wilkinson regression. The FW package implements both the two-steps OLS procedure and Bayesian single step procedure that allows incorporating covariance structure for varieties and environments. We describe the methods implemented in the package and show with examples how this package can be used to perform the Finlay-Wilkinson regression with both methods.

Model Specification and Algorithm

In a reaction norm model (Gregorius and Namkoong, 1986; Perkins and Jinks, 1968) the phenotypic record of the kth replicate of the ith variety observed in the jth environment is modeled as follows

$$y_{\text{ijk}} = \mu + g_{\text{i}} + h_j + b_i h_j + \varepsilon_{ijk} \qquad \text{[Eq. 1]}$$

where g_i is the main effect of ith variety and h_j is the main effect of the jth environment. When we reorganize Eq. 1 into the form: $y_{\text{ijk}} = \mu + g_{\text{i}} + (b_i + 1)h_j + \varepsilon_{ijk}$, we can recognize that $b_i + 1$ is the change of variety performance due to the per unit change of

environment. e_{ijk} is an error term, usually assumed to be IID normal with mean zero and variance s^2_e. If there are no replicates the index *k* can be removed. With this, the equation reduces to $y_{ij} = \mu + g_i + h_j + b_i h_j + \varepsilon_{ij}$. The collection of parameters to be estimated from the model of Eq. 1 include the intercept and the vectors of effects: $\mathbf{g} = \{g_i\}$, $\mathbf{b} = \{b_i\}$ and $\mathbf{h} = \{h_j\}$.

Estimation using two steps methods

The estimation of the regression of each line on environmental means requires regressing, within-line, the observed phenotypes of the line on environmental means. A standard two-steps approach is as follows:

Step 1, estimate the environmental means using a main effect model

$$y_{ijk} = \mu + g_i + h_j + \varepsilon_{ijk} \qquad \text{[Eq. 2]}$$

The above regression yields estimates of environmental means which can be used in the second step to estimate the intercepts and slopes of each line.

Step 2, replace h_j with $\hat{h}_j$ in equation [1] yielding

$$y_{ijk} = \mu + g_i + \hat{h}_j + b_i \hat{h}_j + \varepsilon_{ijk} \qquad \text{[Eq. 3]}$$

The above regression yields estimates of the desired parameters (μ, g_i, b_i).

Both Eq.2 and Eq.3 can be implemented with either ordinary least squares or mixed models.

In the standard Finlay-Wilkinson regression (Finlay and Wilkinson, 1963) the environmental means are computed without considering genotype effects (this amounts to drop the effects of genotypes in Eq. 2) and then, fitting Eq. 3 separately within each line

in the second step. When both steps are implemented using OLS, the estimated environmental effects reduce to the sample mean of each environment.

The FW package implemented a slightly different OLS two-step procedure as the standard one, where instead of regression within each line in separate linear models, we fit Eq. 3 with a single linear model with all the varieties so we can get an estimate of the error variance σ_ϵ^2.

Bayesian approach

Inferences in a Bayesian model are based on the posterior distribution of unknowns given the data: $p(\boldsymbol{\theta}|\mathbf{y}) \propto p(\mathbf{y}|\boldsymbol{\theta})p(\boldsymbol{\theta})$, where $\boldsymbol{\theta}$ represents the collection of the unknowns: μ, **g**, **b**, **h**, and σ_ε^2, $p(\mathbf{y}|\boldsymbol{\theta})$ is the conditional distribution of the data given the parameters and $p(\boldsymbol{\theta})$ is the joint prior distribution assigned to the model uknowns. According to Eq. 1 and assuming IID normal residuals, we have

$$p(\mathbf{y}|\boldsymbol{\theta}) = \prod_{ijk} \mathrm{N}(\mu + g_i + h_j + b_i h_j, \sigma_\varepsilon^2).$$

In the FW package, the prior density is assumed to have the following form: $p(\boldsymbol{\theta}) = p(\sigma_\varepsilon^2)p(\mathbf{g})p(\mathbf{h})p(\mathbf{b})$. The residual variance σ_ε^2 is assigned a scaled-inverse χ^2 distribution: $\sigma_\varepsilon^2 \sim (\nu_\varepsilon, S_\varepsilon^2)$, with degrees of freedom ν_ε (>0) and scale parameter S_ε^2 (>0). The overall mean μ is assigned a flat prior. The prior distributions for **g**, **b**, **h**, are all multivariate Normal: $\mathbf{h} \sim \mathrm{N}(\mathbf{0}, \mathbf{H}\sigma_h^2)$, $\mathbf{g} \sim \mathrm{N}(\mathbf{0}, \mathbf{A}\sigma_g^2)$, $\mathbf{b} \sim \mathrm{N}(\mathbf{0}, \mathbf{A}\sigma_b^2)$, where **H** is a covariance structure describing co-variances between the environmental means (this can be a covariance structure based on spatial information) and , **A** is a covariance structure describing co-variances between levels of the random effects **g** and **b** (**A** could be either a pedigree or marker-derived relationship matrix). Independence between the effects of the levels of any of the random effects can be obtained by setting either **A** or **H** to be an

identity matrix. Since σ_h^2, σ_g^2 and σ_b^2 are also unknown, they are assigned scaled-inverse-χ^2 distributions whose shape are controlled by a set of hyper parameters. The FW package offers users the possibility of specifying hyper parameters (degree of freedom and scale parameters); however, if these are not specified, specific sets of rules similar to those described in (Pérez et al., 2010) are used to determine those parameters. Further details about this are given in the appendix.

In the model described above the posterior density does not have a closed form; however, estimates of features of the posterior distribution (e.g., posterior means, posterior standard deviations, or confidence regions) can be derived using Monte Carlo Methods. The FW package draws samples from the posterior distribution of the model using a Gibbs sampler (Casella and George, 1992; Geman and Geman, 1984) similar to the one described in (Su et al., 2006), details of the implementation of Gibbs sampler are provided in the Supplemental File 1.

Software

The FW package implements both a two-steps OLS procedure and the Bayesian model described in the previous section. Type the following command in R will install the package:

```
library(devtools)
install_github("lian0090/FW")
```

Example (Wheat) data set

The package includes a data set that can be used to run examples. The data set (originally made publicly available by Crossa et al., 2010). contains data for 599 wheat lines from CIMMYT's Global Wheat Program and evaluated for grain yield in four environments. The dataset becomes available in the R environment by running the following R-code:

```
library(FW)
data(wheat)
```

Function `library()` loads the package, and `data()` loads datasets included in the package into the environment. The above code loads the following objects into the environment: - `wheat.Y`, a data.frame (2396 × 3) containing the grain yield of 599 wheat lines in four environments, the three columns are: `VAR` for variety identifiers, `ENV` for environment identifiers, and `y` for grain yield (each entry corresponds to an average of two plot records); -`wheat.G` (599 × 599) is a genomic relationship matrix computed from DArT markers. Further details about this data set can be found in Crossa et al. (2010).

User Interface

All the arguments of the FW function have default values, except the response variable and the corresponding identifiers for varieties and environments. A basic call to the FW program is as follows.

Box 1: Basic Call of the FW Function

```
1  library(FW)
2  data(wheat)
3  attach(wheat.Y)
4  lm1=FW(y=y,VAR=VAR,ENV=ENV,method="OLS")
5  lm2=FW(y=y,VAR=VAR,ENV=ENV)
```

When the call of the FW function is done using the code in line 4 of Box 1, FW fits a Finlay-Wilkinson regression with the two-steps OLS method: `y` (numeric, *n*, NAs are allowed) is the response variable, `VAR` (character, *n*, NAs are not allowed) are the identifiers for the varieties which are treated as labels; `ENV` (character, *n*, NAs are not allowed) are the identifiers for the environments; `method` is used to describe what method to use: "OLS" for ordinary least squares. The default method (`"Gibbs"`) is the Bayesian regression; this can be invoked using the code in line 5 of Box 1. By default, a single chain of Gibbs sampler is run with a total of 5,000 cycles and the samples from the first 3,000 cycles are used for Burn-in, samples from the remaining 2,000 cycles for inference (the user is advised to run longer chains and check convergence as well as the size Monte Carlo errors). The FW function provided many additional arguments; details can be found in the user manual.

After fitting either OLS or Gibbs method, FW function returns a list with estimates and arguments used in the call, a brief description of the outputs follows.

Return

Box 2 shows the structure of the object returned after calling the FW function. The first element `$y` of the list is the response vector used in the call to FW, `$whichNa` gives the position of the entries in `y` that were missing, `$mu` (vector), `$g` (matrix), `$b`

(matrix), `$h` (matrix) are the estimated posterior means of μ, **g**, **b**, **h**; `$yhat` (matrix) is the estimated posterior means of the predictor $\hat{\mathbf{y}}$: $\hat{y}_{ijk} = \hat{\mu} + \hat{g}_i + \hat{h}_j + \hat{b}_i\hat{h}_j$

With OLS method, `$g`, `$b`, `$h` and `$yhat` all have only one column; with Gibbs method each column provides estimates derived from one MCMC chain. Since the default behavior is to run only one chain the outputs in Box 2 contain only one column; however, if multiple chains are run, estimates from different chains are provided in different columns.

The output `$var_e`, `$var_g`, `$var_b`, `$var_h` are the estimated posterior means for σ_ε^2, σ_g^2, σ_b^2 and σ_h^2 (only available for Gibbs method). Each element of `$var_e`, `$var_g`, `$var_b` and `$var_h` correspond to estimates derived from different chains.

Box 2: Structure of the object returned by FW

```
List of 15
 $ y        : num [1:2396] 6.17 3.14 2.74 3.26 4.99 ...
 $ whichNa  : int(0)
 $ VAR      : chr [1:2396] "775" "775" "775" "775" ...
 $ ENV      : chr [1:2396] "1" "2" "4" "5" ...
 $ mu       : Named num 4.65
 $ g        : num [1:599, 1] -0.456 0.177 -0.609...
 $ b        : num [1:599, 1] 0.1542 -0.1392 0.1066 ...
 $ h        : num [1:4, 1] 0.505 -0.198 -0.789 -1.396 ...
 $ yhat     : num [1:2396, 1] 5.15 4.28 3.54 2.79 5.22 ...
 $ var_e    : Named num 0.3
 $ var_g    : Named num 0.0869
 $ var_b    : Named num 0.0952
 $ var_h    : Named num 0.849
```

Output files

No output files are generated for OLS method. For Gibbs method, samples for σ_ε^2, σ_g^2, σ_b^2, σ_h^2, and (by default) the first two elements of **g, b, h** will be saved; as the Gibbs sampler collects samples, these samples are saved to the hard drive (only the most recent

samples are retained in memory); by default, a thinning of 5 is used. Once the iteration process finishes, FW will read all the saved samples into a `mcmc` object, save the `mcmc` object into a file `samps.rda`, and remove the raw sample files. To prevent overloading the RAM with samples by default FW only save samples of the two first entries of the vectors of random effects; however the user can change this behavior by specifying which entries of the vectors are desired using the saveVAR (for **g** and **b**) and saveENV (for **h**) argument. These samples produced by FW can be used to assess convergence and to estimate Monte Carlo error. The file `samps.rda` can be directly loaded into R using `load('samps.rda')`. Once the object containing the samples is loaded in the R environment, the package coda (Plummer et al., 2006) can be used to obtain plots of the chains and compute convergence diagnostics.

Application examples

The FW package can easily fit the FW model with OLS or Gibbs method. It can also easily generate plots for the samples and for the fitted model. Users are offered flexibility over the hyper-parameter setup and output for the Gibbs method. We will demonstrate these features with examples. We present basic examples in the main text of this articles and examples involving fine-tuning the Gibbs method (e.g., hyper-parameter setup, fitting more than two chains, specify saved samples) as Supplementary data.

In Example 1, we illustrated how the package can be used to fit Finlay-Wilkinson regression by OLS method and Gibbs method with and without covariance structure. The second example illustrates how to use FW for cross-validation.

Example 1: Fitting models with default setup for 599 wheat lines

Box 3 shows the code used to fit a FW regression using three different approaches: (i) a two-steps OLS model (code in line 3), (ii) a Bayesian FW regression assuming independence of lines and of environments (code in lines 5-6) and (iii) a Bayesian FW regression that incorporates genomic information (lines 8-9). In the Bayesian models, the seed for the random number generator can be specified using the argument `seed` (see lines 5-9) and the argument `saveAt` can be used to add a path and a pre-fix to be appended to `'samps.rda'` file.

Box 3: Fit models by default parameters

```
1   library(FW); data(wheat); attach(wheat.Y)
2
3   OLS=FW(y=y,VAR=VAR,ENV=ENV, method="OLS")
4
5   GibbsI=FW( y=y,VAR=VAR,ENV=ENV,
6              method="Gibbs",seed=12345,saveAt="GibbsI",nIter=50000
7              ,burnIn=5000)
8
9   GibbsA=FW(y=y,VAR=VAR,ENV=ENV,
10             method="Gibbs",A=wheat.G,seed=12345,
11             saveAt="GibbsA",nIter=50000,burnIn=5000)
12
13  load("GibbsIsamps.rda")
14  HPDinterval(sampsI[,c("var_e","var_g","var_b","var_h")])
15
16  load("GibbsIsamps.rda")
17  HPDinterval(sampsI[,c("var_e","var_g","var_b","var_h")])
```

The parameter estimates can be directly extracted from the `FW` object as illustrated in Box 2. The 95% credible intervals for the parameters can be obtained by the `HPDinterval` function after loading the samples into memory (line 13-17 of Box 3). In table 1, we listed the estimates of variance components from the three models. For OLS method, only the residual variance σ_ε^2 is estimated, and the estimates of σ_ε^2 are very similar across the three models. Also from Table 1, we can see that the variance of the

main effects of the environments is large relative to both the error variance and the phenotypic variance (in this case phenotypes were scaled to a unit variance); therefore, we should expect that most of the prediction accuracies would come from the estimate of environment contrasts.

Table 1. Estimated variance components (posterior 95% credible intervals in parenthesis) from different models

Parameters	FW output	OLS	GibbsI (A=I)	GibbsA (A=G)
σ_ε^2	$var_e	0.32	0.30 (0.28, 0.32)	0.30 (0.28, 0.32)
σ_g^2	$var_g	NA	0.09 (0.07, 0.11)	0.11 (0.08, 0.14)
σ_b^2	$var_b	NA	0.10 (0.07, 0.12)	0.13 (0.10, 0.17)
σ_h^2	$var_h	NA	0.90 (0.24, 1.90)	0.88 (0.24, 1.88)

The fitness of the models can be examined by the correlations between the observed values y and the fitted values $\hat{y}$ (line 1 of Box 4). The OLS model fitted the data better than both GibbsI and GibbsA: the correlation was 0.91 for the OLS method, 0.88 those for GibbsI and 0.86 for GibbsA.

Box 4: Correlation between y and $\hat{y}$, and correlations for b among different models.

```
1  cor(y,OLS$yhat);cor(y,GibbsI$yhat);cor(y,GibbsA$yhat);
2  cor(OLS$b,GibbsI$b);cor(OLS$b,GibbsA$b);cor(GibbsI$b,GibbsA$b)
```

In Table 2, we listed the correlations among parameter estimates from different models (code for parameter b was provided in line 2 of Box 4), and noticed that correlations among parameters estimates from different models are high.

Table 2. Pearson's product-moment correlation between parameter estimates derived by each of the three methods implemented in Box 3.

	OLS-GibbsI	OLS-GibbsA	GibbsI-GibbsA
$\hat{h}$	1.00	1.00	1.00
$\hat{b}$	0.94	0.81	0.83
$\hat{g}$	0.98	0.79	0.81
$\hat{y}$	0.96	0.94	0.97

The pattern of variety performance in different environments can be visualized by plotting the observed and fitted values against the estimated environment effects. Figure 1 was generated by the calling of `plot` function in line 2 of Box 5. By comparing the plot for OLS and GibbsA, we observed that the fitted values for OLS had larger range than that for GibbsA due to the effect of shrinkage in GibbsA. Since there are 599 lines, it is hard to identify each line. The plot function can selectively plot user specified varieties through the argument `plotVAR`. Figure 2 has only five varieties and is produced by the code in line 4-9 of Box 5.

The slope in the plot corresponds to $1 + b_i$ and the dashed grey line corresponds to a slope equals to 1 ($b_i = 0$), as we have already explained in the model specification, the average change of variety performance per unit change of environment effect is $1 + b_i$. We observe from Figure 4 that line ID=1081265 performs well in all environments and line ID=13302 is better adapted to good environments.

Box 5: Plot fitted models

```
1  par(mfrow=c(1,2))
```

```
plot(OLS,main="OLS"); plot(GibbsA,main="GibbsA")

plot(OLS, plotVAR=c("1081265","1101307",
                    "1295736", "13302" , "1343502"), main="OLS")

plot(GibbsA, plotVAR=c("1081265","1101307",
                    "1295736", "13302" , "1343502"),
main="GibbsA")
```

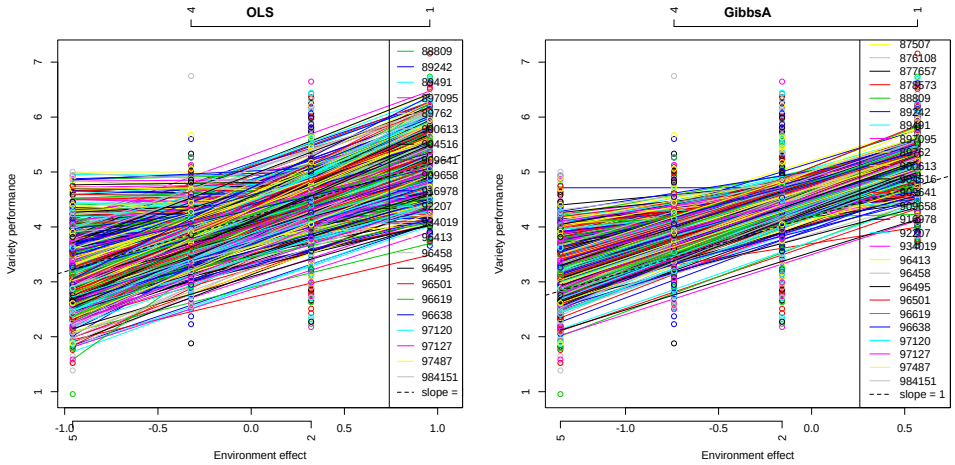

Figure 1. plot of variety performance on estimated environment values. Each color represents a different variety. Lines are fitted values and circles are the cell means of genotype and environment combination.

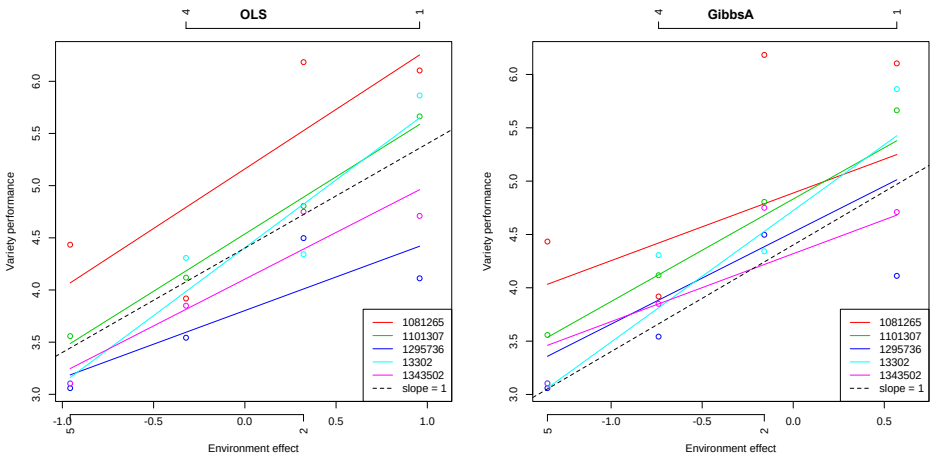

Figure 2. plot of the performance of five varieties on estimated environment values. Each color represents a different variety. Lines are fitted values and circles are the cell means of genotype by environment combination.

Assessment of convergence for Bayesian FW regressions

The convergence of Gibbs sampler can be examined by plotting the samples collected by FW. The code in Box 6 illustrates how to produce trace plots: lines 1-2 load and plot the samples from GibbsI and lines 3-4 do the same for GibbsA. Mixing was reasonably good in both cases for the variance components (σ_ϵ^2 (`var_e`), σ_g^2 (`var_g`), σ_b^2 (`var_b`)) , genotype main effects **g** (`g`), genotype slope **b** (`b`) and the function predictor $\hat{y}$ (`yhat`). There are many high peaks in the trace plot of σ_h^2 (`var_h`), which might be expected since there are only four levels of environment effect, and therefore, we might expect large variances in the estimate of σ_h^2. Figure 3 reproduces the trace plot of the variance components (`var_e, var_g, var_b, var_h`)

The mixing for the intercept μ (`mu`) and the environment effect **h** (`h`) can be relatively poor due to the relatively large effect of **h** (`h`) and therefore the possible confounding among **h** and μ. Figure 4 reproduces the trace plot for intercept μ(`mu`) and

the first two elements of environment effect h_1 (`h[1]`) and h_2(`h[2]`) in GibbsA. Similar confounding effects between μ (`mu`) and **h** (`h`) have been observed by (Shariati et al., 2009) when treating **h** (`h`) as fixed effects and only the contrasts between environment effects are converging well. We therefore suggest the user to run longer chains to get enough samples for estimating μ and **h**.

Box 6: Plot Gibbs samples

```
load("GibbsIsamps.rda")
plot(samps,density=F,ask=T)
load("GibbsAsamps.rda")
plot(samps,density=F,ask=T)
```

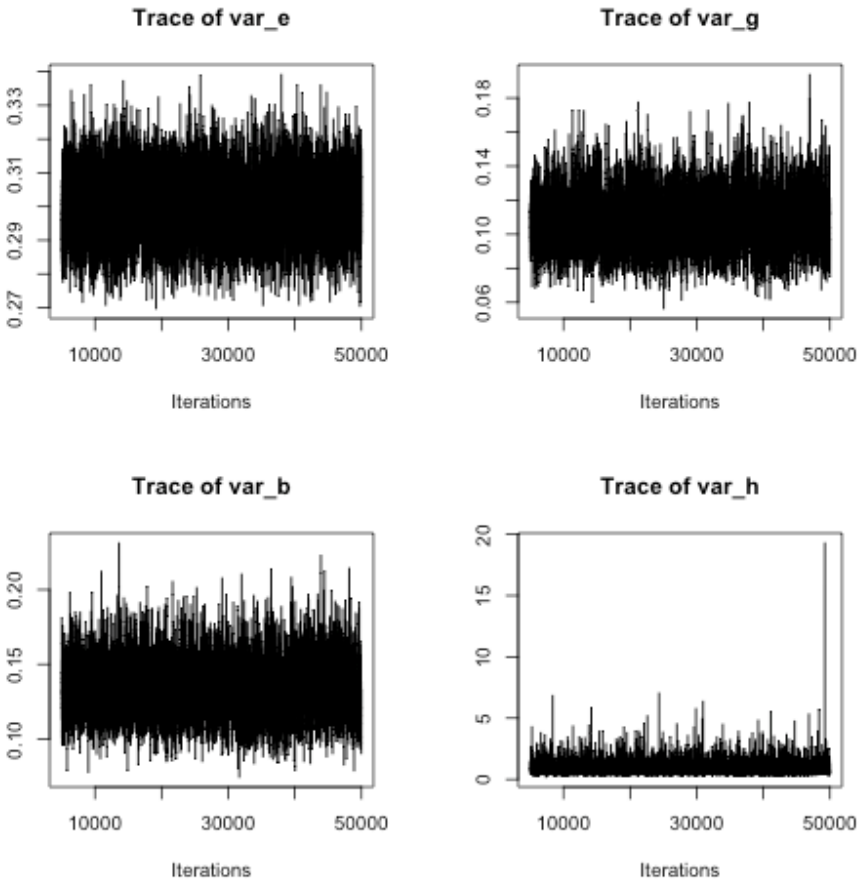

Figure 3. Trace plot of variance components from GibbsA.

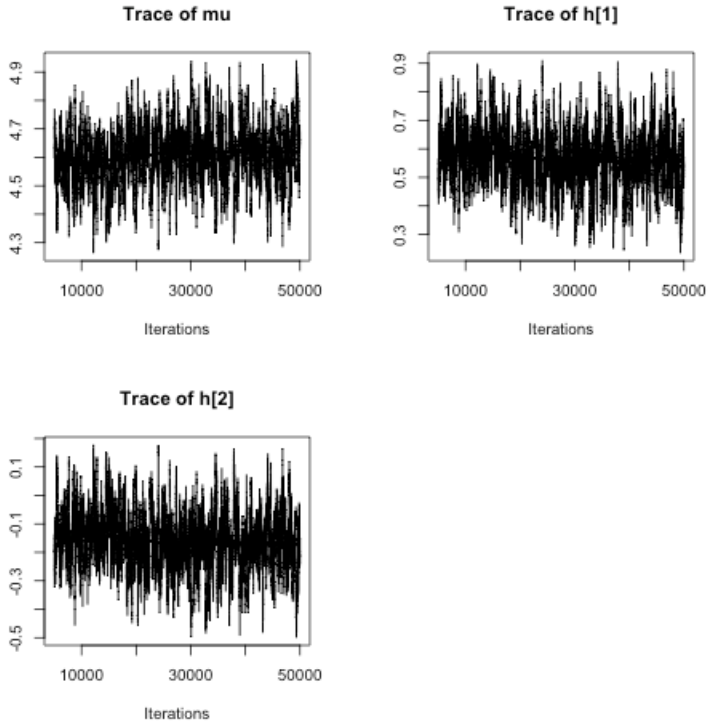

Figure 4. Trace plot of the intercept μ (`mu`) and the first two levels environment effects h_1(`h[1]`) and h_2(`h[2]`).

Example 2: Assessment of prediction accuracy in testing data sets

Example 1 suggests that the OLS method fitted the data better than the Bayesian models. However, better model fitness does not necessarily imply higher prediction accuracy. In the following example we illustrate how to use the FW for assessment of prediction accuracy using cross-validation.

To assess the ability of the model for predicting new data, we first masked some values of `y` as missing values: create a new variable `yNA` which is equal to `y`; for each variety, randomly select one environment as missing and the corresponding values of

`yNA` is set to `NA` (line 1 to line 4 in Box 7). The masked values of `y` are treated as validation set and the remaining values of `y` are treated as the training set.

The same code in Example 1 can be used here to fit the three models except setting `y=yNA` (line 7-8 of Box 7). Correlation between y and $\hat{y}$ were calculated for the training data (line 10-12 of Box 7) and the validation data (line 14-16 of Box 7).

Box 7: correlation between y and $\hat{y}$ for training and validation data sets.

```
1   yNA=y
2   set.seed(12345)
3   whichNa=seq(from=0,to=2392,by=4)+sample(1:4,size=599,replace=T)
4   yNA[whichNa]=NA
5
6   OLS=FW(y=yNA,VAR=VAR,ENV=ENV, method="OLS")
7   GibbsI=FW(y=yNA,VAR=VAR,ENV=ENV,
8   method="Gibbs",seed=12345,nIter=50000, burnIn=5000)
9   GibbsA=FW(y=yNA,VAR=VAR,ENV=ENV,
10  method="Gibbs",A=wheat.G,seed=12345,nIter=50000,burnIn=5000)
11
12  cor(y[-whichNa],OLS$yhat[-whichNa,])
13  cor(y[-whichNa],GibbsI$yhat[-whichNa,])
14  cor(y[-whichNa],GibbsA$yhat[-whichNa,])
15
16  cor(y[whichNa],OLS$yhat[whichNa,])
17  cor(y[whichNa],GibbsI$yhat[whichNa,])
18  cor(y[whichNa],GibbsA$yhat[whichNa,])
```

The $r(y, \hat{y})$ for training data follows the same trend as in Example 1, where OLS fitted the data best: 0.94 for OLS, 0.88 for GibbsI and 0.86 for GibbsA. However, The $r(y, \hat{y})$ for the validation set has reserved orders: 0.62 for OLS, 0.79 for GibbsI and 0.81 for GibbsA. Both GibbsI and GibbsA perform better than OLS in predicting new values.

We also noted in Table 3 that the correlations for the parameter estimates among different models reduced compared to Example 1 due to the missing values. For example,

the correlation for the estimated **b** among different models reduced to 0.85 between OLS and GibbsI, 0.64 between OLS and GibbsA, and 0.79 between GibbsI and GibbsA.

Table 3. Pearson's product-moment correlation between parameter estimates derived by each of the three methods implemented in Box 7.

	OLS-GibbsI	OLS-GibbsA	GibbsI-GibbsA
$\hat{h}$	1.00	1.00	1.00
$\hat{b}$	0.85	0.64	0.79
$\hat{g}$	0.96	0.74	0.77
$\hat{y}$	0.91	0.87	0.97

Computation time for 599 wheat lines

We ran the FW function in an Intel Core i7 1867 MHz Processor (R was executed in a single thread) with 16 GB of RAM memories. We recorded the memory and time usage for Gibbs methods with 50000 iterations. With the full dataset (599 varieties, 2396 observations) the process took around 45 M of RAM memory for GibbsA, 16 M of RAM for GibbsI and 153 M for OLS. The time needed to finish the process was: 11 minutes for GibbsA, 3 minutes for GibbsI and 2 seconds for OLS.

Conclusions

The FW package allows fitting Finlay-Wilkinson regression with ordinary least square method and Bayesian method. For Bayesian method, covariance matrix among varieties and environments can be included in the model. The interface allows the user to fit the models (e.g. OLS versus Gibbs) and visualize the results easily. The algorithms for

Gibbs Sampler are implemented in C and the speed is high. The package also provided flexibility for changing the hyper-parameters and model output.

Acknowledgements

We thank the collaborators in national agricultural research institutes who carried out the Elite Spring Wheat Yield Trials (ESWYT) and provided the phenotypic data analyzed in this article. GDLC and LL received financial support from NIH grants R01GM101219 and R01GM099992 and from Arvalis. GDLC received financial support from Arvalis and CIMMYT.

References

Casella, G., and E. I. George. 1992. Explaining the gibbs sampler. The American Statistician. 46:167-174.

Copas, J. B. 1983. Regression, prediction and shrinkage. Journal Of The Royal Statistical Society. Series B (Methodological). :311-354.

Crossa, J., G. de Los Campos, P. Pérez, D. Gianola, J. Burgueño, J. L. Araus, D. Makumbi, R. P. Singh, S. Dreisigacker, and J. Yan. 2010. Prediction of genetic values of quantitative traits in plant breeding using pedigree and molecular markers. Genetics. 186:713-724.

Finlay, K., and G. Wilkinson. 1963. The analysis of adaptation in a plant-breeding programme. Crop And Pasture Science. 14:742-754.

Frank, L. E., and J. H. Friedman. 1993. A statistical view of some chemometrics regression tools. Technometrics. 35:109-135.

Geman, S., and D. Geman. 1984. Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. Pattern Analysis And Machine Intelligence, Ieee Transactions On. :721-741.

Gregorius, H.-R., and G. Namkoong. 1986. Joint analysis of genotypic and environmental effects. Theoretical And Applied Genetics. 72:413--422.

Perkins, J. M., and J. Jinks. 1968. Environmental and genotype-environmental components of variability. 3. multiple lines and crosses.. Heredity. 23:339--356.

Plummer, M., N. Best, K. Cowles, and K. Vines. 2006. Coda: convergence diagnosis and output analysis for mcmc. R News. 6:7-11.

Pérez, P., G. de Los Campos, J. Crossa, and D. Gianola. 2010. Genomic-enabled prediction based on molecular markers and pedigree using the bayesian linear regression package in r. The Plant Genome. 3:106.

Shariati, M., I. Korsgaard, and D. Sorensen. 2009. Identifiability of parameters and behaviour of mcmc chains: a case study using the reaction norm model. Journal Of Animal Breeding And Genetics. 126:92-102.

Su, G., P. Madsen, M. S. Lund, D. Sorensen, I. R. Korsgaard, and J. Jensen. 2006. Bayesian analysis of the linear reaction norm model with unknown covariates. Journal Of Animal Science. 84:1651-1657.