Chapter 12
Genomic Best Linear Unbiased Prediction

Genomic best linear unbiased prediction (gBLUP) is the most commonly used method for genomic prediction. Although the Bayesian Alphabet series are considered the most popular methods, they are not robust in the sense that they may perform well in one data set but poorly in other data sets. These Bayesian methods require detailed prior distributions and the priors are often selected based on the natures of the data. The gBLUP method, however, uses only simple maximum likelihood method for parameter estimation and requires only the well-known Henderson's mixed model equation to calculate the BLUP for random effects. Therefore, it is the most robust method in the sense that it performs well in most types of data. For some specific data, it may not be optimal, but its performance has never been too low. On average, its performance is better than all other methods.

1. Ridge regression is equivalent to best linear unbiased prediction

The linear model for a centered response variable *y* ($n\times 1$ vector) fit by a scaled matrix of predictors *X* ($n\times m$ matrix) is

$$y = X\beta + \varepsilon \tag{1}$$

where β ($m\times 1$ vector) are the regression coefficients and ε ($n\times 1$ vector) are the residual errors with mean zero and $\text{var}(\varepsilon) = I_n\sigma^2$. Let λ be the ridge parameter. The ridge estimates of the regression coefficients are

$$\beta^{Ridge} = (X^T X + \lambda I_m)^{-1} X^T y \tag{2}$$

The ridge regression problem can be formulated as a Bayesian regression problem by assigning each regression coefficient a prior normal distribution with mean zero and variance $\text{var}(\beta_k) = \sigma_\beta^2 = \sigma^2 / \lambda$ where $\lambda = \sigma^2 / \sigma_\beta^2$. Since *y* is centered, the expectation of *y* is 0 and the variance of *y* is

$$\text{var}(y) = XX^T\sigma_\beta^2 + I_n\sigma^2 = (XX^T + \lambda I_n)\sigma_\beta^2 \tag{3}$$

Given λ, the best linear unbiased predictor of β has two forms, one being the same as the ridge estimate given in equation (2) and the other being derived from the conditional expectation of β given y. Let

$$\text{E}\begin{bmatrix} y \\ \beta \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \tag{4}$$

and

$$\text{var}\begin{bmatrix} y \\ \beta \end{bmatrix} = \begin{bmatrix} XX^T\sigma_\beta^2 + I_n\sigma^2 & X\sigma_\beta^2 \\ X^T\sigma_\beta^2 & I_m\sigma_\beta^2 \end{bmatrix} = \begin{bmatrix} XX^T + \lambda I_n & X \\ X^T & I_m \end{bmatrix}\sigma_\beta^2 \tag{5}$$

The conditional expectation of β given y is

$$\begin{aligned}\mathrm{E}(\beta \mid y) &= \mathrm{E}(\beta) + \mathrm{cov}(\beta, y)\left[\mathrm{var}(y)\right]^{-1}\left[y - \mathrm{E}(y)\right] \\ &= \mathrm{cov}(\beta, y)\left[\mathrm{var}(y)\right]^{-1} y \\ &= X^T(XX^T + \lambda I_n)^{-1} y\end{aligned} \tag{6}$$

This is called the best linear unbiased predictor (BLUP) and denoted by

$$\beta^{Blup} = X^T(XX^T + \lambda I_n)^{-1} y \tag{7}$$

The two forms of estimators of β appear to be quite different, but they are exactly the same, as proved in the next paragraph. However, the computation complexities are different. If $n > m$, β^{Ridge} is more efficient because the estimation involves the inverse of an $m \times m$ matrix with a time complexity of $O(m^3)$. If $n < m$, β^{Blup} is more efficient because the estimation involves the inverse of an $n \times n$ matrix with a time complexity of $O(n^3)$. Let put ridge and BLUP together, we have

$$\beta^{Ridge} = (X^T X + \lambda I_m)^{-1} X^T y$$
$$\beta^{Blup} = X^T(XX^T + \lambda I_n)^{-1} y$$

We now prove that the two forms of estimation are the same. The proof requires the following Woodbury matrix identity,

$$(XX^T + \lambda I_n)^{-1} = \lambda^{-1} I_n - \lambda^{-1} X (X^T X + \lambda I_m)^{-1} X^T \tag{8}$$

The original form of Woodbury matrix identifies are called Sherman-Morrision-Woodbury formulas (Golub and van Loan 1996, p.50), represented by

$$\underbrace{(XHX^T + R)^{-1}}_{n \times n} = R^{-1} - R^{-1} X \underbrace{\left(X^T R^{-1} X + H^{-1}\right)^{-1}}_{q \times q} X^T R^{-1}$$

and

$$|XHX^T + R| \;=\; |R||H||X^T R^{-1} X + H^{-1}|$$

****

Substituting the inverse matrix in β^{Blup} by the above identity, we have

$$\begin{aligned}X^T(XX^T + \lambda I_n)^{-1} y &= \lambda^{-1} X^T y - \lambda^{-1} X^T X (X^T X + \lambda I_m)^{-1} X^T y \\ &= \lambda^{-1}(X^T X + \lambda I_m)(X^T X + \lambda I_m)^{-1} X^T y - \lambda^{-1} X^T X (X^T X + \lambda I_m)^{-1} X^T y \\ &= \left[\lambda^{-1}(X^T X + \lambda I_m) - \lambda^{-1} X^T X\right](X^T X + \lambda I_m)^{-1} X^T y \\ &= \left[\lambda^{-1} X^T X + I_m - \lambda^{-1} X^T X\right](X^T X + \lambda I_m)^{-1} X^T y \\ &= (X^T X + \lambda I_m)^{-1} X^T y\end{aligned} \tag{9}$$

which is the same as β^{Ridge}. Note that we used a very simple trick to complete the proof, that is

$$\lambda^{-1} X^T y = \lambda^{-1} \wedge X^T y = \lambda^{-1}(X^T X + \lambda I_m)(X^T X + \lambda I_m)^{-1} X^T y \tag{10}$$

2. Mixed model and the REML method

Let y be an $n \times 1$ vector for the phenotypic values of *n* individuals and it is described by the following mixed model,

$$y = X\beta + Z\gamma + \varepsilon \tag{1}$$

where *X* is a design matrix for fixed non-genetics effects and β is a vector of the fixed effects (including population structural effects, the intercept and other non-genetics effects), Z and γ are the genotype indicator variables and effects of *m* markers of the entire genome, and ε is a vector of residual errors. In this model, β is treated as fixed effects and γ treated as random effects. Therefore, the model is a linear mixed model. For the random effects, we assume normal distributions as described below,

$$\begin{aligned} \varepsilon &\sim MVN_n(0, I_n\sigma^2) \\ \gamma &\sim MVN_m(0, I_m\phi^2 / m) \end{aligned} \tag{2}$$

where σ^2 is the residual error variance ϕ^2 is the polygenic variance of all *m* markers on the genome. The expectation and variance-covariance matrix of y are

$$\mathrm{E}(y) = X\beta$$

and $\mathrm{var}(y) = V$, where

$$\begin{aligned} V &= \mathrm{var}(Z\gamma) + \mathrm{var}(\varepsilon) \\ &= Z\,\mathrm{var}(\gamma)Z^T + I\sigma^2 \\ &= (ZZ^T / m)\phi^2 + I\sigma^2 \\ &= \left(K\lambda + I\right)\sigma^2 \\ &= H\sigma^2 \end{aligned} \tag{3}$$

where $K = ZZ^T / m$ is a matrix of relatedness among individuals inferred from genome-wide markers (identical-by-state matrix), $H = K\lambda + I$, and $\lambda = \phi^2 / \sigma^2$ is the ratio of the polygenic variance to the residual variance. Note that this ratio is defined differently from the variance ratio in the ridge regression analysis described previously. The K matrix is also called a marker interred kinship matrix, which can also be calculated via summation of individual marker genotype indicators,

$$K = \frac{1}{m}\sum_{k=1}^{m} Z_k Z_k^T \tag{4}$$

There are many different versions of the kinship matrix, which differ mostly in the normalization factors. The general expression may be

$$K = \frac{1}{d}\sum_{k=1}^{m} Z_k Z_k^T \tag{5}$$

where $d = m$ is a special case. The most convenient normalization factor is

$$d = \frac{1}{n}\mathrm{tr}\sum_{k=1}^{m} Z_k Z_k^T \tag{6}$$

where $\mathrm{tr}\,\mathrm{MATRIX}$ is the trace of MATRIX (sum of the diagonal elements of the matrix). The normalization factor defined this way will make sure that the estimated polygenic

variance ϕ^2 is in the same scale as the residual variance σ^2. In the pseudo code expression, we may write

$$K = \sum_{k=1}^{m} Z_k Z_k^T; \quad d = \mathrm{tr}(K); \quad K = \frac{1}{d} K$$

The parameter vector is $\theta = [\beta \quad \lambda \quad \sigma^2]^T$. The data are represented by *y*. Given the parameters, the distribution of *y* is multivariate normal. From the above three things, we can construct the log likelihood function,

$$\begin{aligned} L(\theta) &= -\frac{1}{2}\ln|V| - \frac{1}{2}(y - X\beta)^T V^{-1}(y - X\beta) \\ &= -\frac{1}{2}\ln|H| - \frac{n}{2}\ln(\sigma^2) - \frac{1}{2\sigma^2}(y - X\beta)^T H^{-1}(y - X\beta) \end{aligned} \tag{7}$$

The restricted maximum likelihood (REML) method is often better than the maximum likelihood (ML) method for estimation of variance components. The log likelihood function for REML is

$$\begin{aligned} L_R(\theta) &= -\frac{1}{2}\ln|V| - \frac{1}{2}(y - X\beta)^T V^{-1}(y - X\beta) - \frac{1}{2}\ln|X^T V^{-1} X| \\ &= -\frac{1}{2}\ln|H| - \frac{n}{2}\ln(\sigma^2) - \frac{1}{2\sigma^2}(y - X\beta)^T H^{-1}(y - X\beta) - \frac{1}{2}\ln|X^T H^{-1} X| + \frac{q}{2}\ln(\sigma^2) \end{aligned} \tag{8}$$

where $q = \mathrm{rank}(X)$, i.e., the number of independent columns of matrix *X*. We now present the two log likelihood functions together,

$$\begin{aligned} L(\theta) &= -\frac{1}{2}\ln|V| - \frac{1}{2}(y - X\beta)^T V^{-1}(y - X\beta) \\ L_R(\theta) &= -\frac{1}{2}\ln|V| - \frac{1}{2}(y - X\beta)^T V^{-1}(y - X\beta) - \frac{1}{2}\ln|X^T V^{-1} X| \end{aligned} \tag{9}$$

or

$$\begin{aligned} L(\theta) &= -\frac{1}{2}\ln|H| - \frac{n}{2}\ln(\sigma^2) - \frac{1}{2\sigma^2}(y - X\beta)^T H^{-1}(y - X\beta) \\ L_R(\theta) &= -\frac{1}{2}\ln|H| - \frac{n}{2}\ln(\sigma^2) - \frac{1}{2\sigma^2}(y - X\beta)^T H^{-1}(y - X\beta) - \frac{1}{2}\ln|X^T H^{-1} X| + \frac{q}{2}\ln(\sigma^2) \end{aligned} \tag{10}$$

The difference between the two likelihood functions is obvious. Hereafter, we only discuss the REML method for estimation of variance components. The REML likelihood function can be simplified into

$$L_R(\theta) = -\frac{1}{2}\ln|H| - \frac{1}{2\sigma^2}(y - X\beta)^T H^{-1}(y - X\beta) - \frac{1}{2}\ln|X^T H^{-1} X| - \frac{n-q}{2}\ln(\sigma^2) \tag{11}$$

The REML estimate of parameter vector θ is obtained by maximizing the restricted likelihood function.

2. Profiling the REML likelihood function

Assume that λ is known. We can solve for β and σ^2 by maximizing the restricted log likelihood function. The first partial derivatives of $L_R(\theta)$ with respect to β and σ^2 are

$$\frac{\partial}{\partial \beta} L_R(\theta) = \frac{1}{\sigma^2} X^T H^{-1}(y - X\beta)$$
$$\frac{\partial}{\partial \sigma^2} L_R(\theta) = \frac{1}{2\sigma^4}(y - X\beta)^T H^{-1}(y - X\beta) - \frac{n-q}{2\sigma^2} \quad (12)$$

Let these partial derivatives to zero, we have

$$\begin{aligned} &\frac{1}{\sigma^2} X^T H^{-1}(y - X\beta) = 0 \\ &X^T H^{-1} y - X^T H^{-1} X\beta = 0 \\ &X^T H^{-1} X\beta = X^T H^{-1} y \\ &\beta = (X^T H^{-1} X)^{-1} X^T H^{-1} y \end{aligned} \quad (13)$$

and

$$\begin{aligned} &\frac{1}{2\sigma^4}(y - X\beta)^T H^{-1}(y - X\beta) - \frac{n-q}{2\sigma^2} = 0 \\ &(y - X\beta)^T H^{-1}(y - X\beta) - \sigma^2(n-q) = 0 \\ &\sigma^2(n-q) = (y - X\beta)^T H^{-1}(y - X\beta) \\ &\sigma^2 = \frac{1}{n-q}(y - X\beta)^T H^{-1}(y - X\beta) \end{aligned}$$

Therefore,

$$\begin{aligned} \hat{\beta} &= (X^T H^{-1} X)^{-1} X^T H^{-1} y \\ \hat{\sigma}^2 &= \frac{1}{n-q}(y - X\beta)^T H^{-1}(y - X\beta) \end{aligned}$$

These solutions are explicit and both are function of λ because $H = K\lambda + I$.
Substituting these solutions back to the original restricted likelihood function, we have

$$L_R(\lambda) = -\frac{1}{2}\ln|H| - \frac{1}{2\hat{\sigma}^2}(y - X\hat{\beta})^T H^{-1}(y - X\hat{\beta}) - \frac{1}{2}\ln|X^T H^{-1} X| - \frac{n-q}{2}\ln(\hat{\sigma}^2) \quad (14)$$

which is only a function of λ and thus a simple Newton iteration suffices to find the solution for λ. This likelihood function can be simplified into

$$L_R(\lambda) = -\frac{1}{2}\ln|H| - \frac{1}{2}\ln|X^T H^{-1} X| - \frac{1}{2}(n-q)\ln(\hat{\sigma}^2) - \frac{1}{2}(n-q) \quad (15)$$

Given $\hat{\beta} = (X^T H^{-1} X)^{-1} X^T H^{-1} y$, $\hat{\sigma}^2$ can be expressed as

$$\begin{aligned} \hat{\sigma}^2 &= \frac{1}{n-q}(y - X\beta)^T H^{-1}(y - X\beta) \\ &= \frac{1}{n-q}\left[y^T H^{-1} y - y^T H^{-1} X (X^T H^{-1} X)^{-1} X^T H^{-1} y \right] \\ &= \frac{1}{n-q} y^T P y \end{aligned}$$

where

$$P = H^{-1} - H^{-1} X (X^T H^{-1} X)^{-1} X^T H^{-1}$$

Therefore, the restricted log likelihood function can also be expressed as

$$L_R(\lambda) = -\frac{1}{2}\ln|H| - \frac{1}{2}\ln|X^T H^{-1} X| - \frac{1}{2}(n-q)\ln(y^T P y) + \frac{1}{2}(n-q)\left[\ln(n-q) - 1\right] \quad (16)$$

The last term (in blue) is a constant in the sense that it has nothing to do with λ and thus should be ignored, leading to

$$L_R(\lambda) = -\frac{1}{2}\ln|H| - \frac{1}{2}\ln|X^T H^{-1} X| - \frac{1}{2}(n-q)\ln(y^T P y) \quad (17)$$

The Newton iteration is

$$\lambda^{(t+1)} = \lambda^{(t)} - \left[\frac{\partial^2 L_R(\lambda^{(t)})}{\partial \lambda^2}\right]^{-1}\left[\frac{\partial L_R(\lambda^{(t)})}{\partial \lambda}\right]$$

Once the iteration process converges, we get the MLE of λ, denoted by $\hat{\lambda}$.

2. Eigen-decomposition algorithm for fast computing

The restricted log likelihood function involving inverse and determinant of matrix *H*, which is an $n \times n$ matrix. If the sample size is very large, repeatedly inverting such a large matrix can be very costly. This cost be substantially reduced using the eigen-decomposition algorithm. Recall that the *H* matrix only appears in the forms of $\ln|H|$, $y^T H^{-1} y$, $y^T H^{-1} X$ and $X^T H^{-1} X$, where $H = K\lambda + I$. We can decompose the kinship matrix into

$$K = UDU^T \quad (18)$$

where $D = \mathrm{diag}(\delta_1 \quad \delta_2 \quad \cdots \quad \delta_n)$ is a diagonal matrix containing all the eigenvalues and U is the eigenvector (a $n \times n$ matrix) with a property of $U^T = U^{-1}$, i.e., $UU^T = I$. This allows us to write matrix *H* in the following form,

$$\begin{aligned} H &= \lambda K + I \\ &= \lambda UDU^T + I \\ &= \lambda UDU^T + UU^T \\ &= U(\lambda D + I)U^T \end{aligned} \quad (19)$$

where $\lambda D + I$ is a diagonal matrix, playing a role like weight. The determinant and inverse of matrix *H* is then expressed as

$$\begin{aligned} |H| &= |U(\lambda D + I)U^T| \\ &= |U||\lambda D + I||U^{-1}| \\ &= |U||U^{-1}||\lambda D + I| \\ &= |\lambda D + I| \end{aligned} \quad (20)$$

because $|U||U^{-1}| = |U| \times \frac{1}{|U|} = 1$. The inverse of matrix *H* is expressed as

$$
\begin{aligned}
H^{-1} &= \left[U(\lambda D + I)U^T\right]^{-1} \\
&= (U^T)^{-1}(\lambda D + I)^{-1}U^{-1} \\
&= U(\lambda D + I)^{-1}U^T
\end{aligned}
\tag{21}
$$

The above equivalence is due to the property of $U^{-1} = U^T$ that leads to $(U^T)^{-1} = (U^{-1})^T = (U^T)^T = U$. Note that matrix $\lambda D + I$ is diagonal and thus

$$
\ln|H| = \ln|\lambda D + 1| = \sum_{j=1}^{n} \ln(\lambda\delta_j + 1) \tag{22}
$$

The diagonal structure of $\lambda D + I$ also leads to an easy way to calculate the inverse, which is also diagonal with the *j*th diagonal element being $1/(\lambda\delta_j + 1)$. We now write the log likelihood function as

$$
L_R(\lambda) = -\frac{1}{2}\ln|H| - \frac{1}{2}\ln|X^T H^{-1} X| - \frac{1}{2}(n-q)\ln(y^T P y) \tag{23}
$$

where

$$
y^T P y = y^T H^{-1} y - y^T H^{-1} X (X^T H^{-1} X)^{-1} X^T H^{-1} y
$$

The H^{-1} never occurs alone but in the form of $a^T H^{-1} b$, which is then expressed as

$$
a^T H^{-1} b = a^T U (D\lambda + I)^{-1} U^T b
$$

Define $a^* = U^T a$ and $b^* = U^T b$, we have

$$
a^T H^{-1} b = a^{*T} (D\lambda + I)^{-1} b^*
$$

Let a_j^* and b_j^* be the *j*th rows of the corresponding matrices, the above quadratic form can be obtained through simple summation,

$$
a^T H^{-1} b = \sum_{j=1}^{n} a_j^* (\lambda\delta_j + 1)^{-1} b_j^{*T}
$$

Since matrix inverse and determinant have been replaced by simple summations of *n* terms involving a_j^*, b_j^*, δ_j and parameter λ, evaluation of the log likelihood can be very efficient because $a^* = U^T a$ and $b^* = U^T b$ are only calculated once prior to the maximum likelihood analysis. More computing time can be saved if $m < n$ because $\delta_j = 0$ for all $j > m$ and the eigenvector U is an $m \times n$ matrix. The upper limit of the summation is *m* rather than *n*.

3. Best linear unbiased prediction of random effects

Let review the linear mixed model again

$$
y = X\beta + Z\gamma + \varepsilon \tag{24}
$$

where Z is an $n \times m$ design matrix and γ is an $m \times 1$ vector of marker effects. In genomic selection, we do not estimate the effects of markers; rather, we are interested in $\xi = Z\gamma$ as a whole, which is interpreted as a polygenic effect. The expectation of ξ is zero and the variance is

$$
\mathrm{var}(\xi) = K\phi^2
$$

The polygenic model is

$$y = X\beta + \xi + \varepsilon \tag{25}$$

with expectation $\mathrm{E}(y) = X\beta$ and variance

$$\mathrm{var}(y) = K\phi^2 + I\sigma^2 = (K\lambda + I)\sigma^2 = H\sigma^2$$

Henderson's BLUP equations under the genomic selection model are

$$\begin{bmatrix} X^T X & X^T \\ X & I + K^{-1}/\lambda \end{bmatrix} \begin{bmatrix} \beta \\ \xi \end{bmatrix} = \begin{bmatrix} X^T y \\ y \end{bmatrix}$$

This leads to

$$\begin{bmatrix} \hat{\beta} \\ \hat{\xi} \end{bmatrix} = \begin{bmatrix} X^T X & X^T \\ X & I + K^{-1}/\lambda \end{bmatrix}^{-1} \begin{bmatrix} X^T y \\ y \end{bmatrix}$$

$$= \begin{bmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{bmatrix} \begin{bmatrix} Q_1 \\ Q_2 \end{bmatrix}$$

Therefore, the BLUP of the polygenic effects are

$$\hat{\xi} = C_{21}Q_1 + C_{22}Q_2$$

and the variance-covariance matrix of the BLUP is

$$\mathrm{var}(\hat{\xi}) = C_{22}\sigma^2$$

In Henderson's notation this variance is actually denoted by $\mathrm{var}(\hat{\xi} - \xi)$. When the sample size is large, this BLUP equation is expensive computationally. Therefore, I introduce the following more efficient BLUP equation,

$$\begin{aligned} \hat{\xi} &= \phi K V^{-1}(y - X\hat{\beta}) \\ &= \lambda K H^{-1}(y - X\hat{\beta}) \\ &= \lambda K(\lambda K + I)^{-1}(y - X\hat{\beta}) \\ &= \lambda UDU^T U(\lambda D + I)^{-1}U^T(y - X\hat{\beta}) \\ &= \lambda UD(\lambda D + I)^{-1}U^T(y - X\hat{\beta}) \end{aligned}$$

This BLUP equation is identical to the one from the Henderson's mixed model equation, but inversion of matrix *H* has been avoided because $\lambda D + I$ is a diagonal matrix. Unfortunately, a simple method to calculate the variance matrix is not available. There is an approximate method,

$$\begin{aligned} \mathrm{var}(\hat{\xi}) &\approx \phi^2 K - \phi^2 K(K\phi^2 + I\sigma^2)^{-1} K\phi^2 \\ &= \phi^2 K - \phi^2 K(K\lambda + I)^{-1} K\lambda \\ &= \lambda\left[K - \lambda K(K\lambda + I)^{-1} K \right]\sigma^2 \\ &= \lambda\left[K - \lambda KU(D\lambda + I)^{-1}U^T K \right]\sigma^2 \end{aligned}$$

which does not involve inversion of large matrices. The variances calculated this way are slightly smaller than the actual variances.

4. The HAT method for evaluation of predictability

Let $\xi = y - X\hat{\beta}$ be the "observed" polygenic effect. It is the phenotypic value adjusted by the fixed effects. We often call $\xi = y - X\hat{\beta}$ the centered phenotypic values. The model goodness of fit is defined in two forms. One form is the squared correlation between the "observed" polygenic effects and the estimated polygenic effects,

$$R^2_{\text{FIT-1}} = \left\{ \frac{\sum_{j=1}^{n} (\xi_j - \bar{\xi})(\hat{\xi}_j - \bar{\hat{\xi}})}{\sqrt{\sum_{j=1}^{n} (\xi_j - \bar{\xi})^2 \sum_{j=1}^{n} (\hat{\xi}_j - \bar{\hat{\xi}})^2}} \right\}^2$$

The other form is

$$R^2_{\text{FIT-2}} = 1 - \frac{\sum (\xi_j - \hat{\xi}_j)^2}{\sum (\xi_j - \bar{\xi})^2}$$

The predictability of leave-one-out cross validation approximated by the HAT method is derived from the HAT matrix

$$\begin{aligned} \hat{\xi} &= \lambda UD(\lambda D + I)^{-1} U^T (y - X\hat{\beta}) \\ &= \lambda UD(\lambda D + I)^{-1} U^T \xi \\ &= H^R \xi \end{aligned}$$

where

$$H^R = \lambda UD(\lambda D + I)^{-1} U^T$$

is the HAT matrix for random models. Define the predicted error for individual *j* as

$$\xi_j - \hat{\xi}_j^R = \frac{1}{1 - h_{jj}^2} (\xi_j - \hat{\xi}_j)$$

where $\hat{\xi}_j^R$ is the predicted polygenic effect for the *j*th individual and h_{jj}^2 is the *j*th diagonal element of matrix H^R (the leverage value of observation *j*). The above relationship leads to

$$\hat{\xi}_j^R = \xi_j - \frac{1}{1 - h_{jj}^2} (\xi_j - \hat{\xi}_j)$$

The predictability via the HAT method also has two forms, one being

$$R^2_{\text{HAT-1}} = \left\{ \frac{\sum_{j=1}^{n} (\xi_j - \bar{\xi})(\hat{\xi}_j^R - \bar{\hat{\xi}}^R)}{\sqrt{\sum_{j=1}^{n} (\xi_j - \bar{\xi})^2 \sum_{j=1}^{n} (\hat{\xi}_j^R - \bar{\hat{\xi}}^R)^2}} \right\}^2$$

and the other form being

$$R^2_{\text{HAT-2}} = 1 - \frac{\sum (\xi_j - \hat{\xi}_j^R)^2}{\sum (\xi_j - \bar{\xi})^2}$$

The two forms of predictability are often very close to each other.

5. An example of hybrid prediction

The sample data were taken from the IMF2 population of hybrid rice with 278 hybrids genotyped for 1619 bins. The sample data are stored in three files: “IMF2-Genotypes.csv” for the genotype data, “IMF2-Phenotypes.csv” for the phenotype, and “kk.csv” for the kinship matrix. For genomic selection using gBLUP, the genotype data are only used for calculating the kinship matrix.

The R codes are stored in three files: “kinship.R”, “mixed.R” and mixedHat.R”.

kinship.R

```
This script calculates the kinship matrix

************************************************************
dir<-"C:\\Users\\SHXU\\Dropbox\\My UCR Teaching\\GEN 234\\Lecture
Notes\\gBLUP";
setwd(dir)
gen<-read.csv(file="IMF2-Genotypes.csv")
phe<-read.csv(file="IMF2-Phenotypes.csv")

z<-as.matrix(gen[,-c(1:4)])

kk<-t(z)%*%z
kk<-kk/mean(diag(kk))
write.csv(x=kk,file="kk.csv",row.names=T)

m<-nrow(z)
n<-ncol(z)
kk0<-matrix(0,n,n)
for(k in 1:m){
   kk0<-kk0+z[k,]%*%t(z[k,])
}
kk0<-kk0/mean(diag(kk0))
write.csv(x=kk0,file="kk0.csv",row.names=T)

************************************************************
```

mixed.R

This script defines the mixed() function

```
*****************************************************************
mixed<-function(x,y,kk,cov="qq"){
      if(cov!="qq"){
            kk<-eigen(kk,symmetric=T)
      }
      uu<-kk$vectors
      delta<-kk$values
      x<-t(uu)%*%x
      y<-t(uu)%*%y
      r<-ncol(x)
      func<-function(lambda){
            h<-1/(lambda*delta+1)
            x1<-x*sqrt(h)
            y1<-y*sqrt(h)
            xx<-crossprod(x1)
            xy<-crossprod(x1,y1)
            yy<-crossprod(y1)
            yx<-t(xy)
            dd1<-sum(log(1/h))
            yPy<-yy-yx%*%solve(xx)%*%xy
            dd2<-log(det(xx))
            loglike<--0.5*dd1-0.5*dd2-0.5*(n-r)*log(yPy)
            return(-loglike)
      }
      fixed<-function(lambda){
            h<-1/(lambda*delta+1)
            x1<-x*sqrt(h)
            y1<-y*sqrt(h)
            xx<-crossprod(x1)
            xy<-crossprod(x1,y1)
            yy<-crossprod(y1)
            yx<-t(xy)
            cc<-solve(xx)
            yPy<-yy-yx%*%cc%*%xy
            beta<-drop(cc%*%xy)
            s2<-drop(yPy/(n-r))
            v<-cc*s2
            stderr<-sqrt(diag(v))
            result<-c(beta,stderr,s2)
            return(result)
      }
      par0<-1
      fit<-optim(par=par0,fn=func,method="L-BFGS-B",lower=1e-8,upper=1e8)
      lambda<-fit$par
      conv<-fit$convergence
      fn1<-fit$value
      fn0<-func(1e-8)
      lrt<-2*(fn0-fn1)
      blue<-fixed(lambda)
      beta<-blue[1:r]
      stderr<-blue[(r+1):(2*r)]
      ve<-blue[2*r+1]
      lod<-lrt/4.61
```

```
      if(lrt<1e-8){
            p<-1
      } else {
            p<-0.5*(1-pchisq(lrt,1))
      }
      va<-lambda*ve
      h2<-va/(va+ve)
      par<-data.frame(beta,stderr,va,ve,lambda,h2,conv,fn1,fn0,lrt,lod,p)
      return(par)
}

**************************************************************************
```

mixedHat.R

This script calls the mixed() function and perform genomic prediction

```
*********************************************************************
dir<-"C:\\Users\\SHXU\\Lecture Notes\\gBLUP";
setwd(dir)
gen<-read.csv(file="IMF2-Genotypes.csv")
phe<-read.csv(file="IMF2-Phenotypes.csv")
kk<-read.csv(file="kk.csv",header=T)

source(file="mixed.R")

kk<-as.matrix(kk[,-1])
n<-nrow(kk)

qq<-eigen(kk,symmetric=T)
uu<-qq$vectors
delta<-qq$values

y<-as.matrix(phe$kgw)
x<-matrix(1,n,1)

fit<-mixed(x=x,y=y,kk=qq,cov="qq")

write.csv(x=fit,file="result.csv",)

lambda<-fit$lambda[1]
h2<-lambda/(1+lambda)
beta<-as.matrix(fit$beta)
rnk<-which(delta>1e-10)
delta<-delta[rnk]
vector<-uu[,rnk]
r<-y-x%*%beta
d<-delta/(delta*lambda+1)
mat<-t(vector)*sqrt(d)
H<-lambda*crossprod(mat)
eHat<-r-H%*%r
xi<-r-eHat
SSE<-sum(eHat^2)
SST<-var(r)*(n-1)
GOOD2<-1-SSE/SST
GOOD1<-cor(r,xi)^2

PRESS<-0
rr<-NULL
gg<-NULL
indxx<-NULL
foldid<-phe$foldid
nfold<-max(foldid)
for(i in 1:nfold){
   indx<-which(foldid==i)
   nk<-length(indx)
   Hkk<-H[indx,indx]
   e<-solve(diag(nk)-Hkk)%*%as.matrix(eHat[indx])
```

```
   PRESS<-PRESS+drop(t(e)%*%e)
   rr<-c(rr,r[indx])
   gg<-c(gg,r[indx]-e)
   indxx<-c(indxx,indx)

}
PRED2<-1-PRESS/SST
PRED1<-cor(rr,gg)^2

R2<-data.frame(nfold,SSE,SST,PRESS,GOOD1,GOOD2,PRED1,PRED2)

write.csv(x=R2,file="R2.csv",row.names=F)

pred<-data.frame(indxx,gg)
newPred<-pred[order(indxx),]
PRED<-cbind(y,r,xi,newPred)
names(PRED)<-c("y","g_obs","g_est","xx","g_pred")
PRED<-PRED[,names(PRED)!="xx"]

write.csv(x=PRED,file="PRED.csv",row.names=F)
**********************************************************************
```

Results of the sample data

The estimated parameters are

β	$s_{\hat{\beta}}$	ϕ^2	σ^2	λ	$h^2 = \phi^2 / (\phi^2 + \sigma^2)$
24.63164	0.140453	3.013101	0.801078	3.761309	0.789974

The predictability measured in various different ways are

SSE	SST	PRESS	FIT-1	FIT-2	HAT-1	HAT-2
143.7805	1023.645	345.7439	0.867779	0.859541	0.662271	0.662243

The model goodness of fit are 0.8677 (FIT-1) and 0.8595 (FIT-2) respectively for the two forms (very close to each other). The predictability measured in the two forms are 0.6623 (HAT-1) and 0.6622 (HAT-2), also very much the same.