

Genomic Prediction in Livestock

Monday May 11, 2015 – Friday May 15, 2015

8:30 AM – 5 PM daily

Course website: qtl.rocks

Preamble – installing Julia

- a. **An overview as to the promise of genomic selection**
Include basic idea of linkage disequilibrium (LD)
- b. **An introduction to simple linear models and the simulation of data for such models**
(using Julia)
Concept of a Model Equation
Other aspects of the model
 - Expected Values, location parameters or First Moments
 - Second Moments or variance-covariance
 - Distributional Assumptions
 - Simulate X
 - Simulate b
 - Simulate e
 - Construct $y=Xb+e$
 - Form a function to simulate data
- c. **The theory and application of Least Squares** (using Julia) **to simulated data**
Ordinary Least Squares
 - Estimating the fixed effects
 - Standard error of fixed effects
 - Estimating linear functions of fixed effects
 - Estimability – is a function able to be estimated
 - Residual standard error
 - Model sum of squares (reductions)
 - Coefficient of Determination
 - Generalized Least Squares and Weighted Least Squares
- d. **An introduction to Monte Carlo methods, including Markov chains (MCMC) via Metropolis-Hastings and Gibbs Sampling**
Integration of a pdf – for example to determine intensity of selection
Numerical integration – Monte Carlo sampling to estimate intensity of selection
More complex example – intensity of selection in a multivariate context
Metropolis-Hastings sampling from a bivariate normal distribution
Gibbs sampling from a bivariate normal distribution
- e. **Application of MCMC (Gibbs sampling) for statistical inference from linear regression** (using Julia)
Livestock Production paper

Genomic Selection in Livestock

Dorian Garrick
Rohan Fernando
Jack Dekkers

May 11 - 15, 2015
Iowa State University

Animal Breeding & Genetics

ANIMAL SCIENCE

Breeding and Genetics Graduate Education Online

USDA NIFA

<http://www.enbgeo.soe.vt.edu/>

Genomic Selection in Livestock

Some housekeeping

Course hours:
8:30 – 12 AM with 30 min. break at ~ 10 AM
Lunch on your own
1:00 – ~5 PM with 30 min. break at ~ 3 PM

Course notes:
Distributed daily + posted at: [gll.rocks](#)

Course social: Tuesday @ 5:30 - details to follow

Field trip: Saturday @ 6 AM - details to follow

Genomic Selection in Livestock

Short course - focus

- Statistical, quantitative genetic, and computational aspects of genomic selection

Next week's Short course - focus

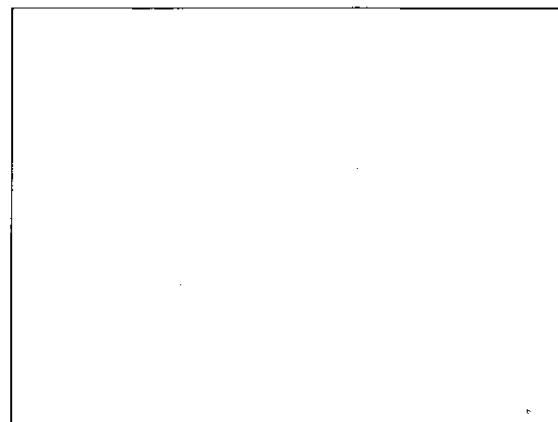
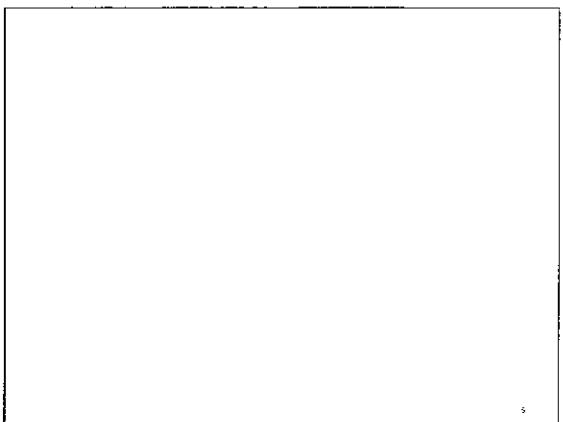
Design of Breeding Programs with Genomic Selection

- Strategies for implementation of genomic selection in livestock breeding programs

Course Outline / Topics

Preamble – installing Julia

- a. Introduction to Genomic Prediction
- b. An introduction to simple linear models and simulation of data for such models
- c. The theory and application of Least Squares (using Julia) to simulated data
- d. Introduction to Monte Carlo methods
- e. Application of MCMC for statistical inference from linear regression
- f. Theory and application of pedigree-based mixed linear models to predict BV
- g. Introduction to Bayes theorem with applications to Bayesian linear regression for genomic analyses
- h. Mixed models fitting marker effects or fitting BV using genomic relationships
- i. The Bayesian alphabet for genomic analyses
- j. GWAS and QTL inference using the Bayesian alphabet
- k. Concepts of estimability and upper limits on accuracy of BayesC0/GBLUP
- l. Imputation, fitting haplotypes and using imputed sequence for GWAS
- m. Single step GBLUP, Single step hybrid models
- n. Multi-trait genomic prediction
- o. Industry applications of genomic prediction



Genomic Prediction Workshop - Ames 2015

Introduction to Genomic Prediction

Dorian Garrick

Lush Endowed Chair in Animal Breeding & Genetics
dorian@iastate.edu

Genomics

gc•no•mics [ˌdʒẽnəmɪks, -mɪkɪs]

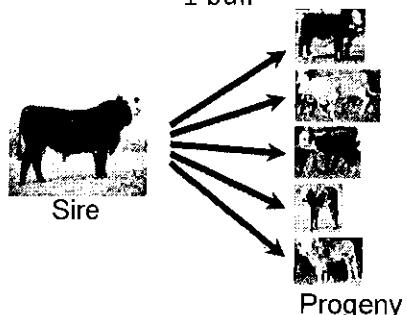
plural noun [treated as sing.]
the branch of molecular biology concerned with the structure, function, evolution, and mapping of genomes.

ORIGIN 1980s: from *genome* 'the complete set of genes present in an organism' + *-ics*.

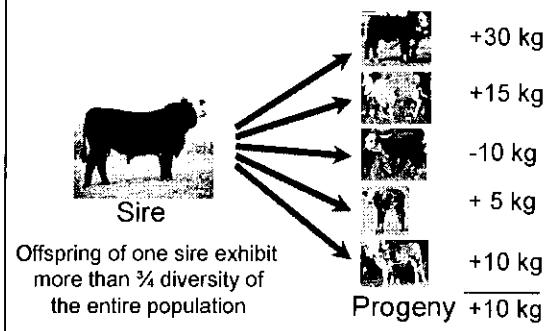
Genomic Prediction

- Ranking candidates for selection using knowledge of the "complete set of genes" *along with conventional pedigree and performance information*
 - Using everything we've got to obtain the most accurate EPD/EBV (at as young an age as possible)

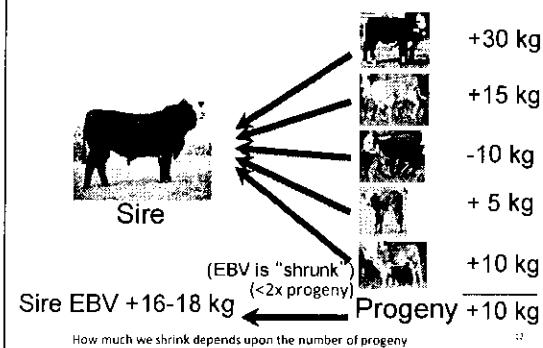
Suppose we generate 100 progeny on 1 bull



Performance of the Progeny



We Learn about Parents from Progeny



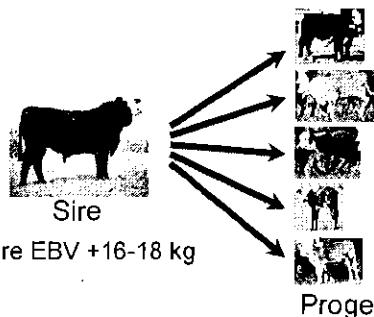
EBVs on widely-used old sires are accurate



With enough progeny,
this is usually close to
the bulls true EBV/EPD
(not surprisingly!)

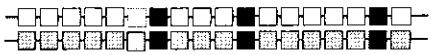
Sire EBV +16-18 kg

Suppose we generate new progeny



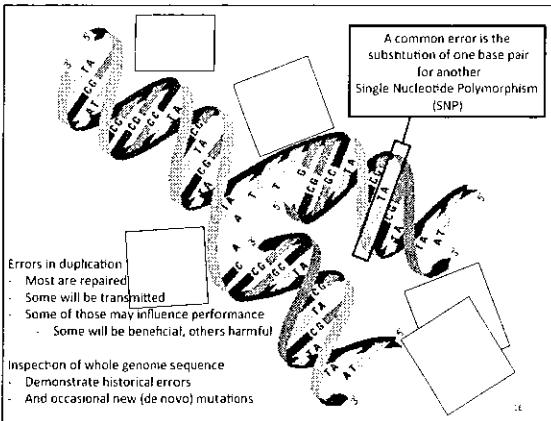
Chromosomes are a sequence of base pairs

Part of 1 pair
of chromosomes



Cattle usually have 30 pairs of chromosomes
One member of each pair inherited from the sire, one from the dam
Each chromosome has about 100 million base pairs (A, G, T or C)
About 3 billion describe the animal

■ Blue base pairs represent genes
■ Yellow represents the strand inherited from the sire
■ Orange represents the strand inherited from the dam

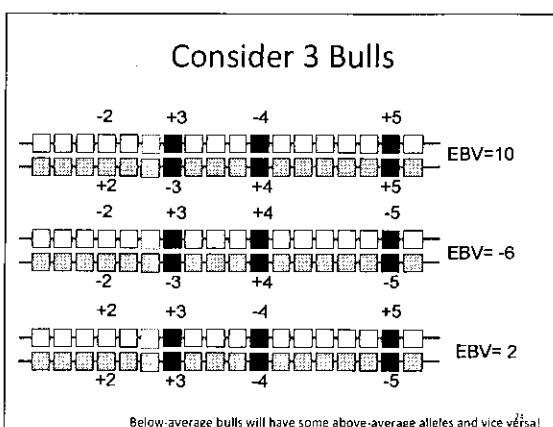
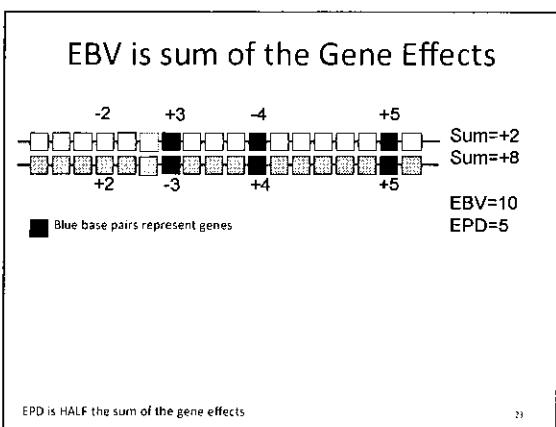
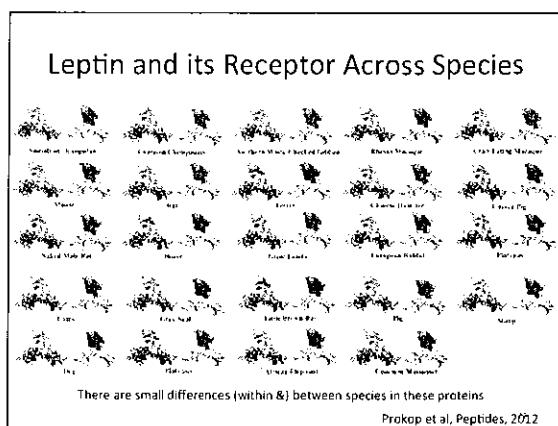
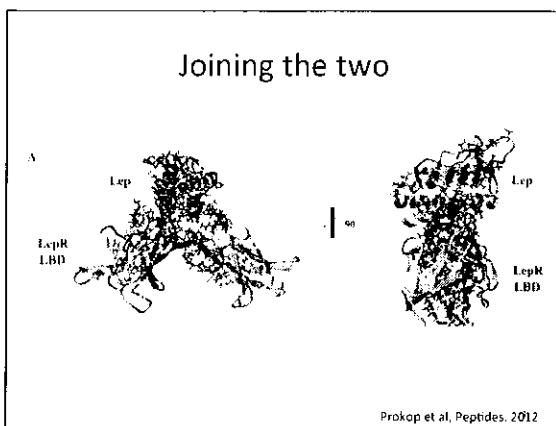
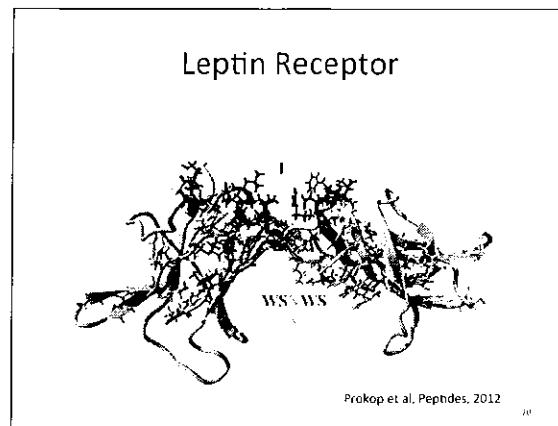
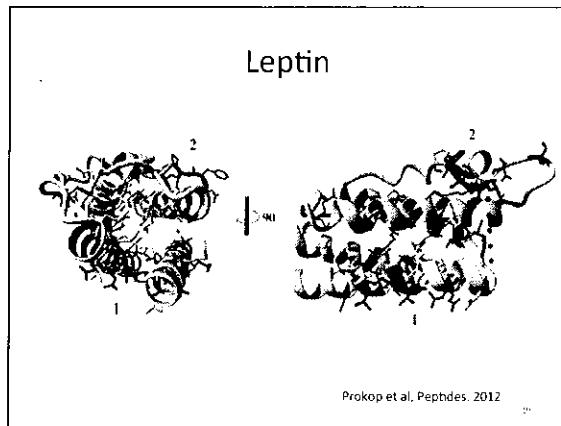


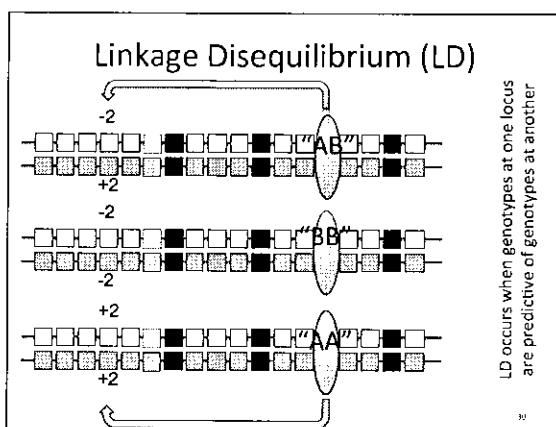
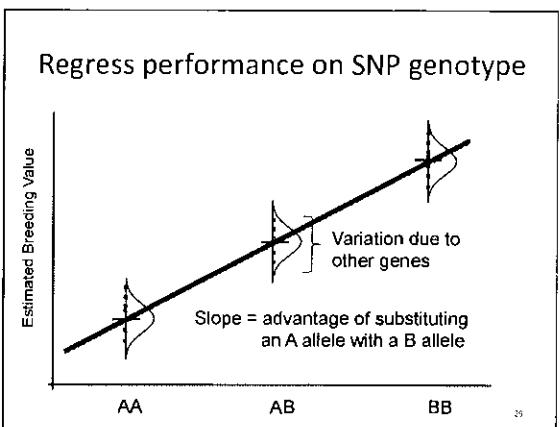
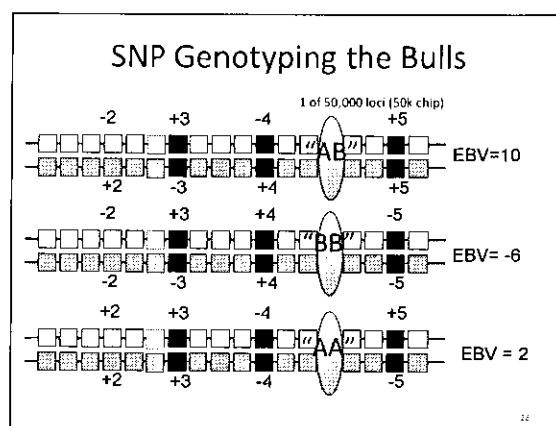
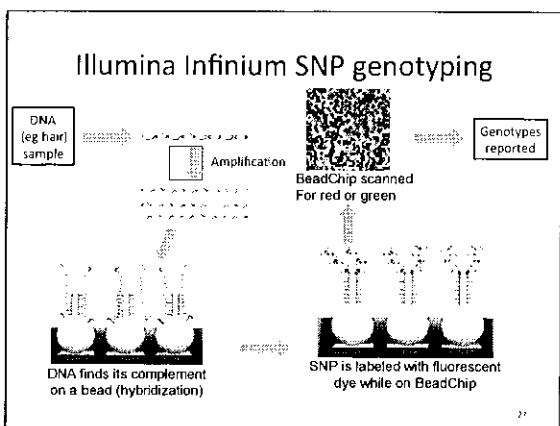
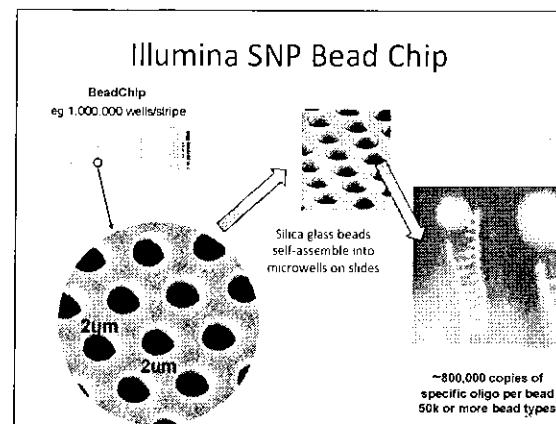
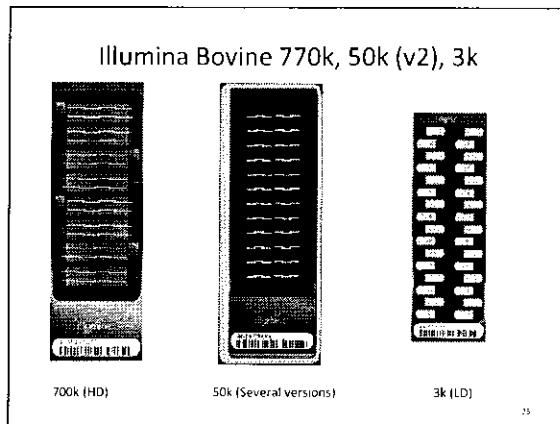
Mutations

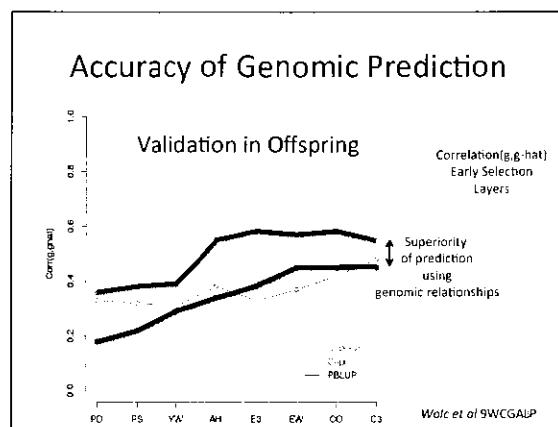
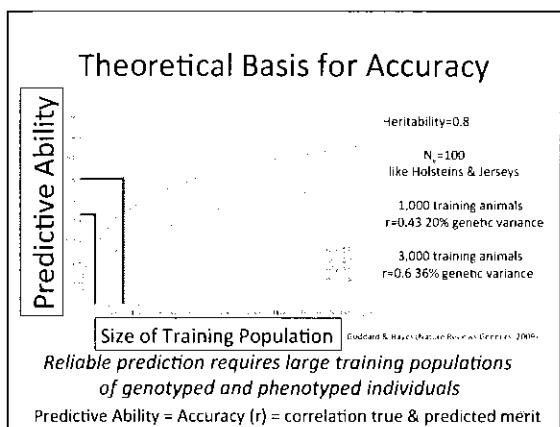
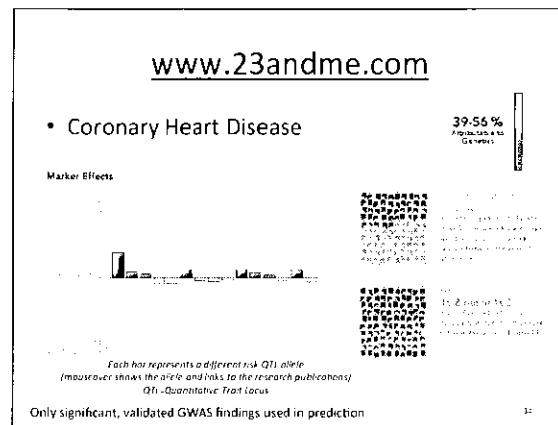
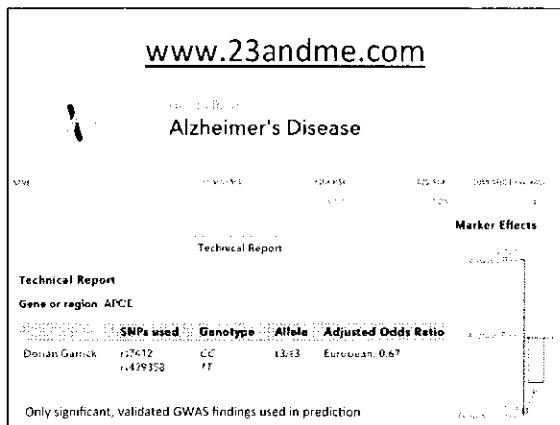
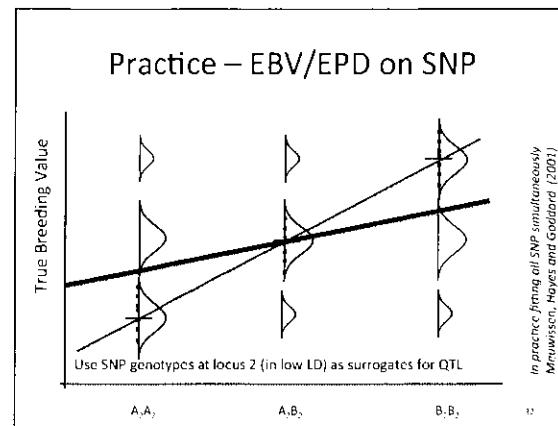
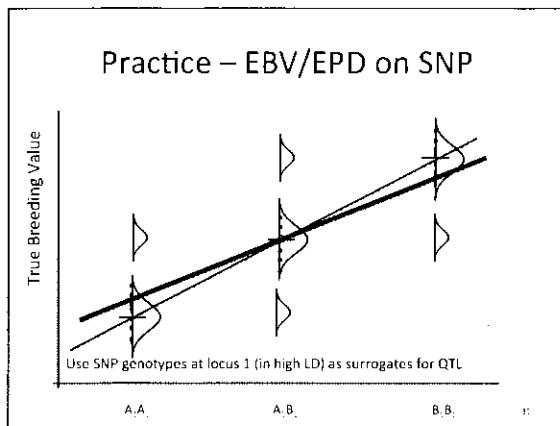
- Could cause complete loss-of-function of the gene (ie the gene is "broken")
 - These can sometimes be catastrophic when an individual is homozygous and carries 2 copies of the broken gene
 - For examples DUMPS, Citrullinemia, BLAD, etc

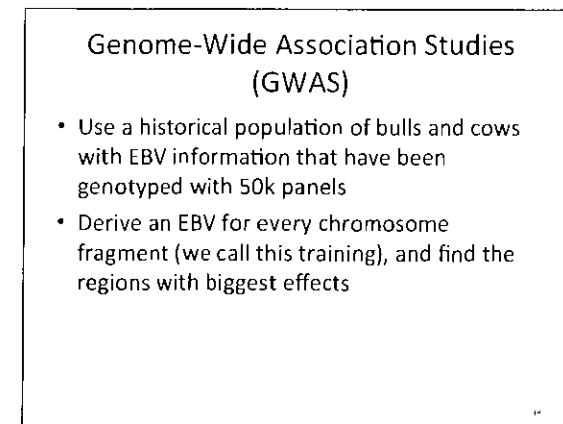
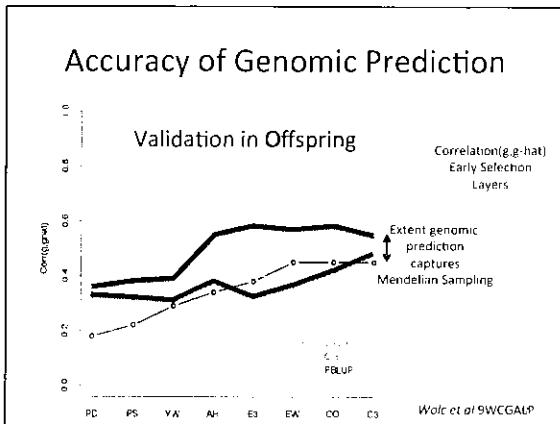
Mutations

- Could cause complete loss-of-function of the gene (ie the gene is "broken")
- Could increase or decrease expression level
- The variant might change amino acid sequence to cause subtle changes to the shape of the protein products making them function a little better or a little worse
 - Natural or artificial selection will favour the variants that improve fitness in that particular climatic and environmental circumstance





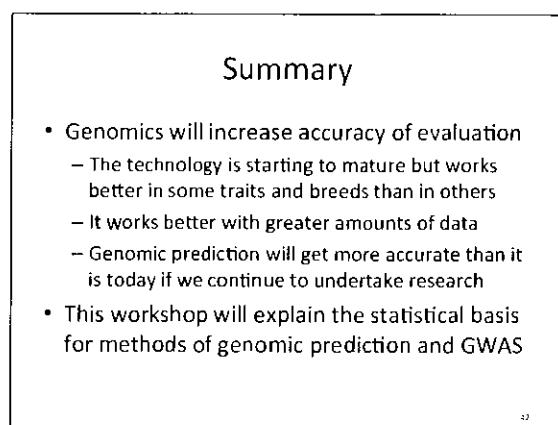
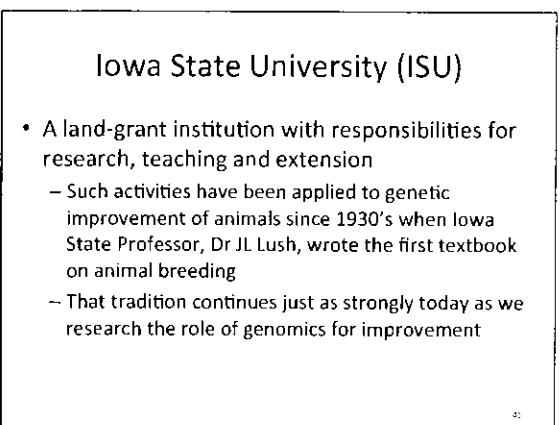
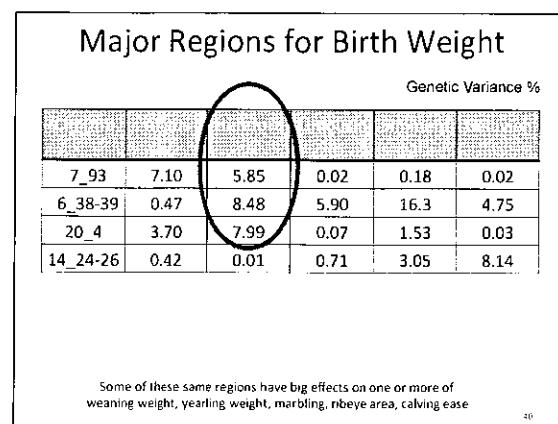




Cut genome into 2,700 1Mb windows

#SNPs	%Var	Cum%Var	map_pos	Regions with biggest effects
11	7.10	7.10	7_93	
28	3.70	10.80	20_4	
22	1.34	12.14	13_58	
22	1.23	13.37	26_34	
9	0.92	14.29	6_29	
25	0.89	16.09	4_75	
26	0.79	16.88	4_114	
23	0.65	17.53	2_121	
17	0.61	18.14	18_55	
25	0.60	18.74	8_88	

Argus Birth Weight



An Introduction to Linear Models

Models

- Concept of a Model Equation
- Other aspects of the model
 - Expected values, location parameters or first moments
 - Second moments or variance-covariance
 - Distributional assumptions

Simple Models

- Performance = Breeding + Feeding
- Phenotype = Genotype + Environment
- Animal Model – model equation

$$y = \text{herd} - \text{year} - \text{season} + BV + e$$

$$y = Xb + Zu + e$$

The “usual” Animal Model

$$\begin{aligned} y &= Xb + Zu + e \\ E[u] &= 0 \text{ and } E[e] = 0 \\ \text{therefore } E[y] &= Xb \end{aligned} \quad \left. \begin{array}{l} 1. \text{ Model Equation} \\ 2. \text{ Location Parameters} \end{array} \right\}$$

$$\begin{aligned} \text{var}[u] &= G = A\sigma_e^2 \quad \text{var}[e] = R = I\sigma_e^2 \quad \text{cov}[u, e] = 0 \\ \text{var}[y] &= V = ZGZ' + R \end{aligned} \quad \left. \begin{array}{l} 3. \text{ Dispersion Parameters} \end{array} \right\}$$

$$y \sim MVN[Xb, V] \quad \left. \begin{array}{l} 4. \text{ Distributional Assumptions} \end{array} \right\}$$

Fixed Effects – Linear Regression

$$y = Xb + e$$

$$E[u] = 0$$

$$\text{var}[e] = R = I\sigma_e^2$$

Perhaps assume $e \sim N[0, I\sigma_e^2]$

$$e \stackrel{\text{iid}}{\sim} N[0, \sigma_e^2]$$

Simple Linear Regression

$$y = Xb + e$$

$$b = \begin{bmatrix} \alpha \\ \beta \end{bmatrix} = \begin{bmatrix} \text{intercept} \\ \text{slope} \end{bmatrix}$$

$$X = \begin{bmatrix} 1 & x_1 \\ 1 & x_2 \\ \vdots & \vdots \\ 1 & x_n \end{bmatrix}$$

Multiple Linear Regression

$$y = Xb + e$$

$$b = \begin{bmatrix} \alpha \\ \beta_1 \\ \vdots \\ \beta_n \end{bmatrix} = \begin{bmatrix} \text{intercept} \\ \text{slope} \\ \vdots \\ \text{slope} \end{bmatrix}$$

$$X = \begin{bmatrix} 1 & x_1 & x_2 & \cdots & x_n \\ 1 & x_1 & x_2 & \cdots & x_n \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_1 & x_2 & \cdots & x_n \end{bmatrix}$$

Estimation

If

$$y = Xb + e$$

then

$$K'y = KXb + Ke$$

for example, choosing $K = X'$

$$X'y = X'Xb + X'e$$

and if $X'y = X'Xb$ then $X'e = 0$

so b is solution to $X'Xb = X'y$

Linear Regression

- Linear Regression
 $y = Xb + e$
- Residual
 $e = y - Xb$, with $E[e] = 0$, and $\text{var}[e] = \sigma_e^2$
- Residual Sum of Squares
 $e'e = (y - Xb)'(y - Xb)$
 $= y'y - y'Xb - b'X'y + b'X'Xb$

Least Squares

- Residual Sum of Squares
 $e'e = y'y - y'Xb - b'X'y + b'X'Xb$
- Take derivatives with respect to vector b
 $de'/db = -X'y - X'y + (X'X + (X'X)')b$
set=0 and solve to find minima/maxima gives
 $X'Xb = X'y$
known as the Least Squares Equations
or the Normal Equations

Estimation

$$\hat{b} \text{ is solution to } X'Xb = X'y$$

which for full rank X is $\hat{b} = [X'X]^{-1}X'y$

$$\begin{aligned} E[\hat{b}] &= E[[X'X]^{-1}X'y] \\ &= [X'X]^{-1}X'E[y] \\ &= [X'X]^{-1}X'Xb = b \\ \text{var}[\hat{b}] &= \text{var}[[X'X]^{-1}X'y] \\ &= [X'X]^{-1}X'\text{var}[y]X[X'X]^{-1} \\ &= [X'X]^{-1}X'\sigma^2 X[X'X]^{-1} \\ &= [X'X]^{-1}X'X[X'X]^{-1}\sigma^2 \\ &= [X'X]^{-1}\sigma^2 \end{aligned}$$

Linear functions of b

$$k'b \text{ is estimated from } k'\hat{b}$$

with $\text{var}[k'\hat{b}] = k'[X'X]^{-1}k\sigma^2$

X not full rank

$k'b$ is estimated from $k'\hat{b}$
with $\text{var}[k'\hat{b}] = k'[X'X] k\sigma^2$
provided $k' = k'[X'X] X'X$

rows of k' can be slacked in a matrix K
 Kb is estimated from $K\hat{b}$
with $\text{var} - \text{cov}[K\hat{b}] = K[X'X] K\sigma^2$
provided $K = K[X'X] X'X$

Residual Standard Error

$$\begin{aligned}\widehat{\sigma} &= MS_{\text{Error}} = SS_{\text{Error}}/df \\ &= (y - X\hat{b})'(y - X\hat{b})/(N - \text{rank}(X)) \\ SS_{\text{Error}} &= SS_{\text{Total}} - SS_{\text{Model}} \\ &= y'y - \hat{b}'X'y \\ R^* &= SS_{\text{Model_mean}}/SS_{\text{Total_mean}} \\ SS_{\text{Model_mean}} &= SS_{\text{Model}} - SS_{\text{Mean}} \\ SS_{\text{Mean}} &= \bar{y}'\bar{y} \\ SS_{\text{Total_mean}} &= SS_{\text{Total}} - SS_{\text{Mean}} \\ &= y'y - N\bar{y}'\end{aligned}$$

Generalized Least Squares

$$\begin{aligned}y &= Xb + (Zu + e) \\ &= Xb + \epsilon \\ \text{var}[y] &= V = ZGZ' + R \\ \hat{b} &\text{ is solution to } X'V^{-1}Xb = X'V^{-1}y\end{aligned}$$

Weighted Least Squares

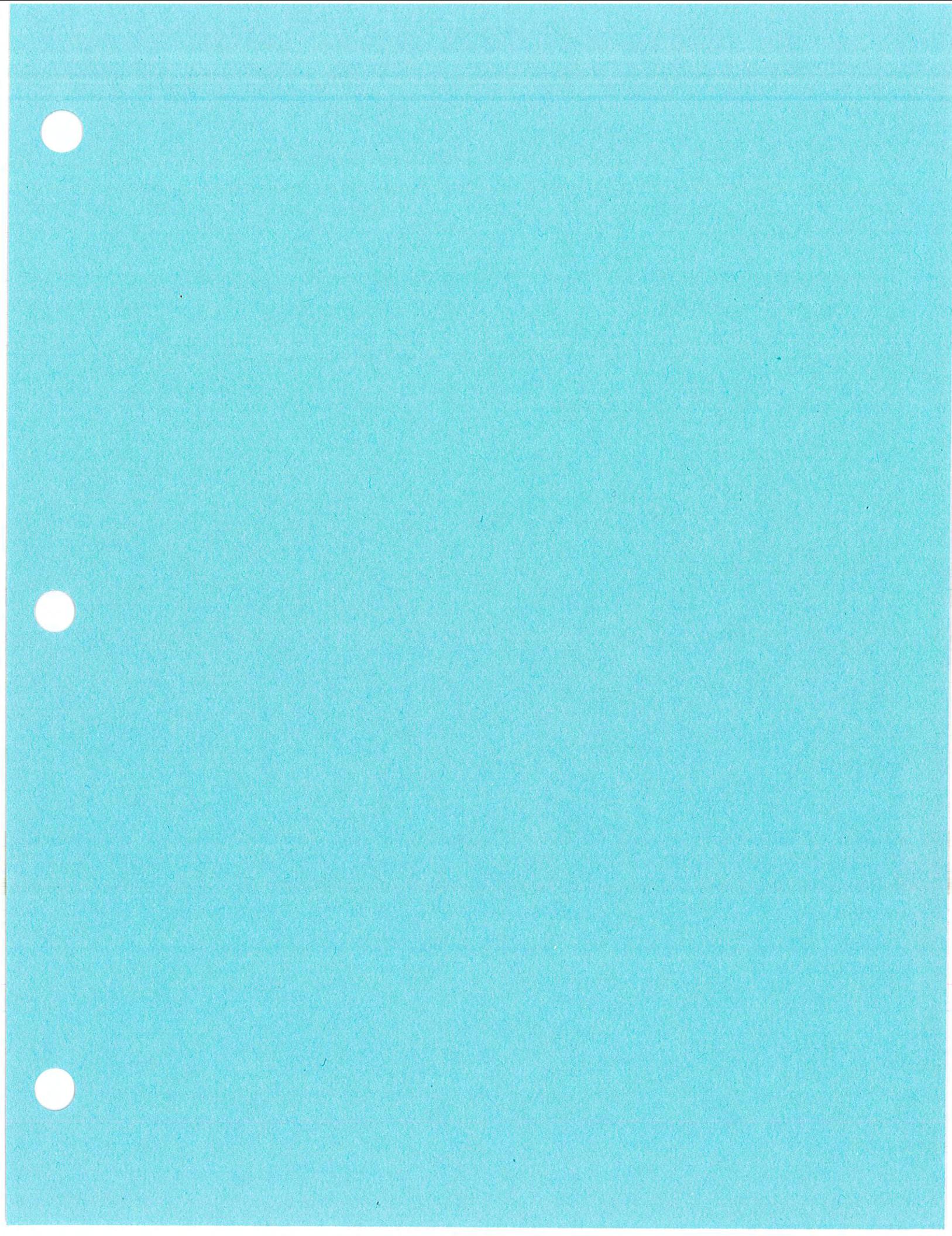
$$\begin{aligned}y &= Xb + e \\ \text{var}[e] &= R = D = \text{diag}(\sigma_e^2) \\ \hat{b} &\text{ is solution to } X'D^{-1}Xb = X'D^{-1}y\end{aligned}$$

Hypothesis Testing

- To test hypotheses we need to know the distribution of the test statistic
 - Which is derived from the distribution of the residuals
 - Commonly assumed to be normally (iid) distributed

Linear Regression

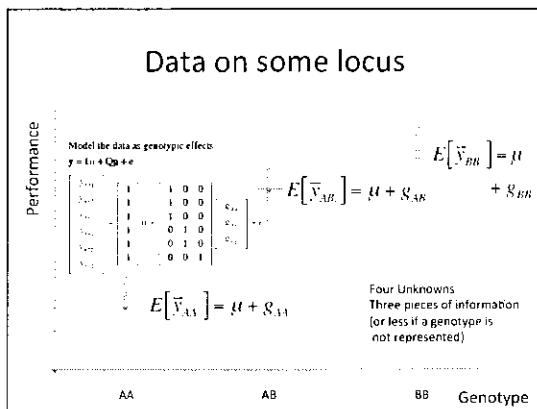
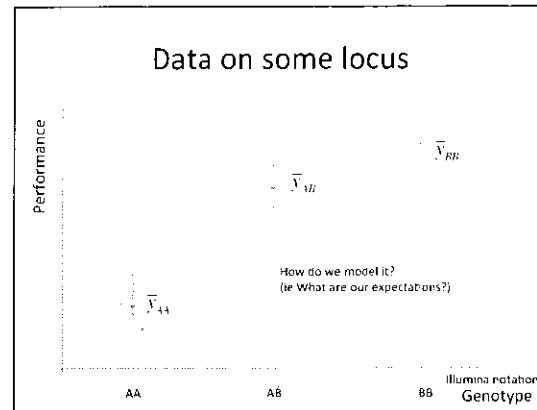
- Least Squares simple linear regression (unknown β_0 and β_1)
- Gibbs Sampler with known σ_e^2
- Bayesian Gibbs sampler with unknown σ_e^2
- As above but with random not fixed β_1
- Bayesian (multiple) linear regression (many random β 's)
- Various models (BLUP, BayesA, B, C, Cr etc)



Look at the back

5/12/15

Fixed effects models
to predict SNP effects



- Parameters and Information Content
- The information content (in fixed effects model) is partly reflected in the degrees of freedom
 - Some degrees of freedom are available to estimate functions of fitted parameters
 - The remainder, if any, contribute to the error sum of squares
 - Overparameterized models have more parameters than (independent) estimable functions

Fixed Effects Model for Genotypes

$$\mathbf{y} = \mathbf{X}\mathbf{b} + \mathbf{W}\mathbf{q} + \mathbf{e}$$

b contains the usual fixed effects

$$\mathbf{q} = \begin{bmatrix} q_{AA} \\ q_{AB} \\ q_{BB} \end{bmatrix}, \text{ defines a class effect}$$

\mathbf{W} is the incidence matrix for AA, AB, BB genotypes and has 3 columns – one for each genotype class and N rows – one for each animal with exactly one 1 in each row according to the genotype of the animal

Fixed Effects Model for Genotypes

$$\mathbf{y} = \mathbf{X}\mathbf{b} + \mathbf{W}\mathbf{q} + \mathbf{e}$$

$$\mathbf{E}[\mathbf{y}] = \mathbf{X}\mathbf{b} + \mathbf{W}\mathbf{q}$$

$$\mathbf{var}[\mathbf{y}] = \mathbf{var}[\mathbf{e}] = \mathbf{I}\sigma_e^2$$

Least Squares Equations

$$\begin{bmatrix} \mathbf{X}'\mathbf{X} & \mathbf{X}'\mathbf{W} \\ \mathbf{W}'\mathbf{X} & \mathbf{W}'\mathbf{W} \end{bmatrix} \begin{bmatrix} \hat{\mathbf{b}} \\ \hat{\mathbf{q}} \end{bmatrix} = \begin{bmatrix} \mathbf{X}'\mathbf{y} \\ \mathbf{W}'\mathbf{y} \end{bmatrix}$$

For $[\mathbf{b}] = [\mu]$, $\mathbf{X} = \mathbf{I}$ In this example
Only fixed effect is mean

$$LHS = \begin{bmatrix} N & n_{AA} & n_{AB} & n_{BB} \\ n_{AA} & n_{AA} & 0 & 0 \\ n_{AB} & 0 & n_{AB} & 0 \\ n_{BB} & 0 & 0 & n_{BB} \end{bmatrix} \quad RHS = \begin{bmatrix} y_{AA} \\ y_{AB} \\ y_{BB} \\ y_{BB} \end{bmatrix}$$

In general equations have order equal to number of fixed effects plus genotypes

No unique solution

$$LHS = \begin{bmatrix} N & n_{AA} & n_{AB} & n_{BB} \\ n_{AA} & n_{AA} & 0 & 0 \\ n_{AB} & 0 & n_{AB} & 0 \\ n_{BB} & 0 & 0 & n_{BB} \end{bmatrix} \quad RHS = \begin{bmatrix} y_{AA} \\ y_{AB} \\ y_{BB} \\ y_{BB} \end{bmatrix}$$

$$\hat{\mathbf{b}} = \begin{bmatrix} 0 \\ \mu + q_{AA} \\ \mu + q_{AB} \\ \mu + q_{BB} \end{bmatrix}, \text{ is one possible solution}$$

No unique solution

$$\hat{\mathbf{b}} = \begin{bmatrix} \mu + q_{BB} \\ q_{AA} - q_{BB} \\ q_{AB} - q_{BB} \\ 0 \end{bmatrix}, \text{ is another possible solution}$$

$$LHS = \begin{bmatrix} N & n_{AA} & n_{AB} & n_{BB} \\ n_{AA} & n_{AA} & 0 & 0 \\ n_{AB} & 0 & n_{AB} & 0 \\ n_{BB} & 0 & 0 & n_{BB} \end{bmatrix} \quad RHS = \begin{bmatrix} y_{AA} \\ y_{AB} \\ y_{BB} \\ y_{BB} \end{bmatrix}$$

Different Solutions have same Estimable Functions

$$\hat{\mathbf{b}}_1 = \begin{bmatrix} \mu + q_{BB} \\ q_{AA} - q_{BB} \\ q_{AB} - q_{BB} \\ 0 \end{bmatrix} \quad \hat{\mathbf{b}}_2 = \begin{bmatrix} 0 \\ \mu + q_{AA} \\ \mu + q_{AB} \\ \mu + q_{BB} \end{bmatrix}$$

Interesting contrasts

$$\mathbf{k}' = [1 \ 1 \ 0 \ 0] \text{ then } \mathbf{k}'\hat{\mathbf{b}}_1 = \mathbf{k}'\hat{\mathbf{b}}_2 = \mu + q_{AA}$$

$$\mathbf{k}' = [0 \ 1 \ -1 \ 0] \text{ then } \mathbf{k}'\hat{\mathbf{b}}_1 = \mathbf{k}'\hat{\mathbf{b}}_2 = q_{AA} - q_{AB}$$

Estimable Functions

- In fixed effects models, many model parameters or functions of model parameters are not estimable, even though a numeric value can be obtained by solving the least squares equations (eg by generalized inverse)

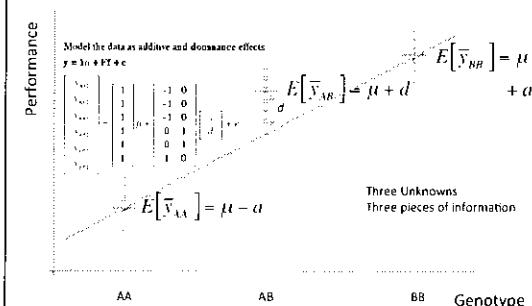
$[\mathbf{X}'\mathbf{X}]^{-1}$ is any generalized inverse of $\mathbf{X}'\mathbf{X}$ if $(\mathbf{X}'\mathbf{X})[\mathbf{X}'\mathbf{X}]^{-1}(\mathbf{X}'\mathbf{X}) = \mathbf{X}'\mathbf{X}$

Define $\mathbf{H} = [\mathbf{X}'\mathbf{X}]^{-1}(\mathbf{X}'\mathbf{X})$

A linear function $\mathbf{k}'\mathbf{b}^0$ is estimable if $\mathbf{k}'\mathbf{H} = \mathbf{k}'$

$\text{var}(\mathbf{k}'\mathbf{b}^0) = \mathbf{k}'[\mathbf{X}'\mathbf{X}]^{-1}\mathbf{k}$ {or $\mathbf{k}'[\mathbf{X}'\mathbf{X}]^{-1}\mathbf{k}\sigma^2$ (if \mathbf{R} was not explicitly fitted)}

Data on some locus



Genotypic vs genetic effects

$\mathbf{g} = \begin{bmatrix} g_{AA} \\ g_{AB} \\ g_{BB} \end{bmatrix}$, genotypic class effects $\mathbf{a} = \begin{bmatrix} -d \\ d \\ 0 \end{bmatrix}$, additive and dominance effects

$$d = \frac{g_{BB} - g_{AA}}{2}, \text{ and } d = g_{AB} - \frac{g_{AA} + g_{BB}}{2}$$

$$\mathbf{K} = \begin{bmatrix} \mathbf{k}_1 \\ \mathbf{k}_2 \end{bmatrix} = \begin{bmatrix} -1 & 0 & 1 \\ 2 & 2 & 2 \\ -1 & 1 & -1 \\ 2 & 2 & 2 \end{bmatrix}, \mathbf{K}\mathbf{q} = \mathbf{a}, \text{ rows of } \mathbf{K} \text{ are orthogonal, } \mathbf{k}_1\mathbf{k}_2 = 0$$

but note \mathbf{g} itself is not estimable, but functions like $g_{kk} - g_{AA}$ are

Equivalent Models

Model	Intercept	Allele A	Allele B	Total
AA	$\mu + g_{AA}$	10	$\mu + d$	$10+13=3$
AB	$\mu + g_{AB}$	14	$\mu + d$	$14+13=1$
BB	$\mu + g_{BB}$	16	$\mu + d$	$16+13=3$

$\mu=0$ $\mu=10$ $\mu=16$ $\mu=13$
 $g_{AA}=10$ $g_{AB}=0$ $g_{BB}=-6$ $\alpha=3$
 $g_{AB}=14$ $g_{AB}=4$ $g_{BB}=-2$ $d=1$
 $g_{BB}=16$ $g_{AB}=6$ $g_{BB}=0$

Both models have the same expectation
Both models have the same variance
Therefore the models are equivalent
(I can fit either model and migrate from one to the other)

Suppose I ignore dominance ($d=0$)

Model the data as an intercept and allele dosage
 $y = \alpha + \beta_1 + \beta_2 + e$

$E[\bar{y}_{AB}] = \alpha + 2\beta$
 $E[\bar{y}_{BB}] = \alpha + \beta$
 $E[\bar{y}_{AA}] = \alpha + 0\beta$
Represents lack of linear fit

Suppose I ignore dominance ($d=0$)

Model the data as a main and substitution effect
 $y = \mu + \tau + e$

$E[\bar{y}_{AB}] = \mu + \tau$
 $E[\bar{y}_{BB}] = \mu$
 $E[\bar{y}_{AA}] = \mu$
Represents lack of linear fit

Suppose I ignore dominance ($d=0$)

Model the data as an intercept and allele dosage
 $y = \alpha + \beta_1 + \beta_2 + e$

$E[\bar{y}_{AB}] = 0\beta_1 + 2\beta_2$
 $E[\bar{y}_{BB}] = 1\beta_1 + 1\beta_2$
 $E[\bar{y}_{AA}] = 2\beta_1 + 0\beta_2$
Represents lack of linear fit

Equivalent Models

Model	Intercept	Allele A	Allele B	Total
AA	$\alpha + 0\beta$	10	$\mu - \tau$	$10 + 2\beta_1 + 0\beta_2 = 10 + 2x5$
AB	$\alpha + 1\beta$	13	μ	$13 + 1\beta_1 + 1\beta_2 = 13 + 5 + 8$
BB	$\alpha + 2\beta$	16	$\mu + \tau$	$16 + 0\beta_1 + 2\beta_2 = 16 + 2x8$

$\alpha=10$ $\mu=13$ $\beta_1=5$
 $\beta_2=8$ $\tau=3$ $NB \beta_1 + \beta_2 = 3$

All models have the same expectation
All models have the same variance
Therefore the models are equivalent
(I can fit any of the models and migrate from one to the other)

Summary Fixed Effects Models

	dominance	d=0	dominance	d=0	d=0
Model df	3	2			
Genotypic	yes	no			
All alleles	yes	yes			
Substitution	yes	yes			
Animals	n/a	n/a			

↔
Equivalent models

Summary Fixed Effects Models

	dominance	d=0	dominance	d=0	d=0
Model df	3	2			
Genotypic	yes	no			
All alleles	yes	yes	yes		
Substitution	yes	yes	yes		
Animals	n/a	n/a			

↔
Equivalent models

↔
Non equivalent models

Fitting SNPs as random effects

Fixed or Random

- Reasonable to consider animal effects as random in the usual context
 - Variation in alleles (ie genotype) between animals that contributes to the genetic variance
 - Not variation in allelic value at a particular locus
- Not so clear that an individual locus (or every loci) should be treated as random
 - Especially when the genotypes are observed and treated as known in the incidence matrix

Suppose we have many loci

The obvious solution is to fit the a effects jointly for every locus

$$\mathbf{y} = \mathbf{X}\mathbf{b} + \mathbf{M}\mathbf{a} + \mathbf{e}$$

$$= \mathbf{X}\mathbf{b} + \sum_{i=1}^{i=n\text{markers}} \mathbf{m}_i a_i + \mathbf{e}$$

a_i is the substitution effect for the i th locus

Singular Coefficient Matrix

- The incidence matrix of genotypes, \mathbf{M} , has n rows (= number of genotyped animals) and p columns (= number of loci/markers/haplotypes)
- Typically using Illumina livestock chips (cattle, horses, pigs, sheep, chickens, dogs) $n < 10,000$ and $p > 40,000$
- If no 2 animals have the same p genotypes, then \mathbf{M} has full row rank
- The $\mathbf{M}'\mathbf{M}$ component of the coefficient matrix cannot be full rank (rank $\mathbf{M}'\mathbf{M}$ is $n < p$)
 - Rank(AB) is at most the lesser of rank(A) and rank(B)

Practical Consequence

- It is not possible using ordinary least squares to simultaneously estimate more than n effects of loci plus other fixed effects
 - Can use stepwise approaches to successively add loci and determine a subset of markers that are informative in the training data
 - But least squares tend to produce upwards biased estimates of effects (especially when power is limiting)
 - Cannot use all markers to predict genomic merit

Alternative Approaches

- Modifications to Least Squares
 - Ridge Regression, Partial Least Squares etc
- Treat a effects as random rather than fixed
 - We routinely fit single and multi-trait animal models with many more effects than observations
 - Provides opportunities for many mixed model procedures, such as BLUP, REML, Bayesian analyses
 - These methods will also “shrink” estimates

Random locus effects

- Following the treatment of locus effects as fixed, we could consider the following possible models for random locus effects
 - A) fitting every genotype at a locus
 - This would require us to describe the variance-covariance matrix between the alternative genotypes
 - That matrix is singular in the absence of dominance
 - B) fitting every allele at a locus
 - C) fitting substitution effect at each locus

and the corresponding partitions of the inverse are

$$\begin{bmatrix} \mathbf{X}'\mathbf{R}^{-1}\mathbf{X} & \mathbf{X}'\mathbf{R}^{-1}\mathbf{Z} \\ \mathbf{Z}'\mathbf{R}^{-1}\mathbf{X} & \mathbf{Z}'\mathbf{R}^{-1}\mathbf{Z} + \mathbf{G}^{-1} \end{bmatrix}^{-1} = \begin{bmatrix} \mathbf{C}^{11} & \mathbf{C}^{12} \\ \mathbf{C}^{21} & \mathbf{C}^{22} \end{bmatrix}$$

In relation to random effects, we need only concern ourselves with the \mathbf{C}^{22} partition of the inverse coefficient matrix. Note however that the entire coefficient matrix must be inverted to obtain the partition of interest. From this partition you have the prediction error variance-covariance matrix. That is,

$$\text{var}[\mathbf{u} - \hat{\mathbf{u}}] = \mathbf{C}^{22}$$

$$\text{var}[\hat{\mathbf{u}}] = \mathbf{G} - \mathbf{C}^{22}, \text{ and recall that } \text{var}[\mathbf{u}] = \mathbf{G}.$$

A common unitfree measure of how well we have estimated the BLUP is the square of the correlation between the true and estimated effect. Since the true effects are not known, this cannot be calculated directly, but is a function of the \mathbf{G} and \mathbf{C}^{22}

matrices. Specifically, $r^2 = \frac{\text{var}[\hat{\mathbf{u}}]}{\text{var}[\mathbf{u}]} = \frac{\text{diag}[\mathbf{G} - \mathbf{C}^{22}]}{\text{diag}[\mathbf{G}]}$ for best linear predictions (BLP)

and best linear unbiased predictions (BLUP).

Exercise 4

In many circumstances we are interested in linear combinations of random effects. For example, we might want to know the BLUP and the r^2 of a team of sires rather than an individual. Alternatively, we might be interested in the contrast or

difference between one or more alternative sires or teams. To compute these, we need to construct a relevant vector of contrasts that we will denote as \mathbf{k} . For

example, to predict the superiority of sire 1 over sire 2, for $\mathbf{u}' = [u_1 \ u_2 \ u_3 \ u_4]$, we would form $\mathbf{k}' = [1 \ -1 \ 0 \ 0]$. To compare a team of the first two sires to

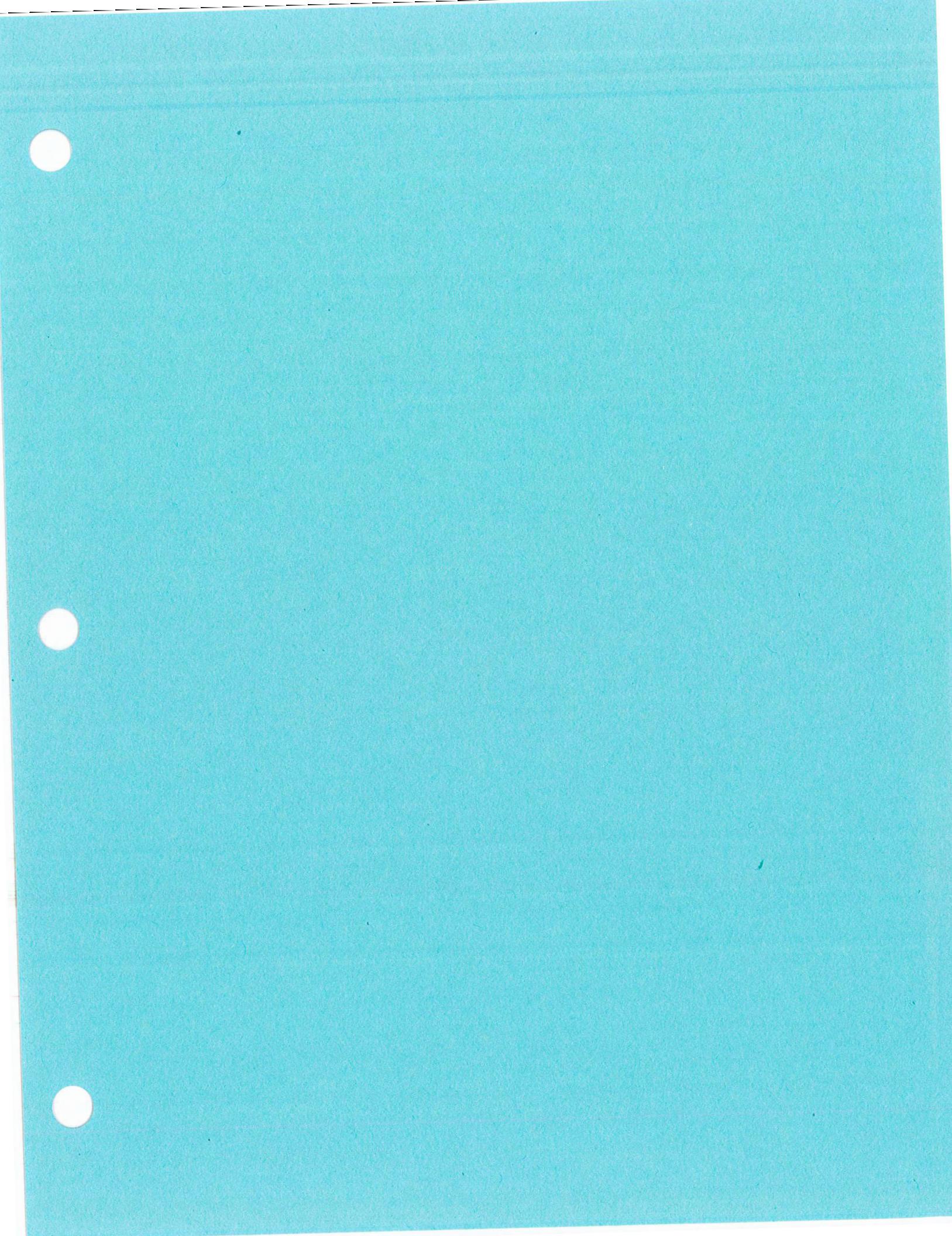
the second two sires we would use $\mathbf{k}' = [0.5 \ 0.5 \ -0.5 \ -0.5]$. Both of these contrasts can be considered simultaneously by stacking them up the rows of \mathbf{k}'

together in a matrix, $\mathbf{K}' = \begin{bmatrix} 1 & -1 & 0 & 0 \\ 0.5 & 0.5 & -0.5 & -0.5 \end{bmatrix}$

The BLUP of $\mathbf{k}'\mathbf{u}$ is simply obtained as $\mathbf{k}'\hat{\mathbf{u}}$, and $\text{var}(\mathbf{k}'\mathbf{u}) = \mathbf{k}'\mathbf{G}\mathbf{k}$,

$$\text{var}(\mathbf{k}'\hat{\mathbf{u}}) = \mathbf{k}'[\mathbf{G} - \mathbf{C}^{22}]\mathbf{k}.$$

Construct some linear combinations, and estimate the prediction error variance and r^2 for these linear combinations.



Introduction to Monte-Carlo Methods

Rohan L. Fernando

May 2015

Mean and Variance of Truncated Normal

Suppose $Y \sim N(\mu_Y, V_Y)$.

The mean and variance of Y given truncation selection are:

$$E(Y|Y > t) = \mu_Y + V_Y^{1/2}i$$

where

$$i = \frac{f(s)}{p}$$

$f(s)$ is the standard normal density function

$$s = \frac{t - \mu_Y}{V_Y^{1/2}}$$

$$p = \Pr(Y > t)$$

$$\text{Var}(Y|Y > t) = V_Y[1 - i(i - s)]$$

Proof:

Start with mean and variance for a standard normal variable given truncation selection.

Let $Z \sim N(0, 1)$.

The density function of Z is:

$$f(z) = \sqrt{\frac{1}{2\pi}} e^{-\frac{1}{2}z^2}$$

The density function for Z given truncation selection is

$$f(z|z > s) = f(z)/p$$

From the definition of the mean:

$$\begin{aligned}
E(Z|Z > s) &= \frac{1}{p} \int_s^\infty zf(z)dz \\
&= \frac{1}{p} [-f(z)]_s^\infty \\
&= \frac{f(s)}{p} \\
&= i
\end{aligned}$$

because the first derivative of $f(z)$ with respect to z is:

$$\begin{aligned}
\frac{d}{dz}f(z) &= \sqrt{\frac{1}{2\pi}} e^{-\frac{1}{2}z^2}(-z) \\
&= -zf(z)
\end{aligned}$$

Now, to compute the variance of Z given selection, consider the following identity:

$$\begin{aligned}
\frac{d}{dz}zf(z) &= f(z) + z\frac{d}{dz}f(z) \\
&= f(z) - z^2f(z)
\end{aligned}$$

Integrating both sides from s to ∞ gives

$$zf(z)]_s^\infty = \int_s^\infty f(z)dz - \int_s^\infty z^2f(z)dz$$

Upon rearranging this gives:

$$\begin{aligned}
\int_s^\infty z^2f(z)dz &= \int_s^\infty f(z)dz - zf(z)]_s^\infty \\
\frac{1}{p} \int_s^\infty z^2f(z)dz &= \frac{1}{p} \int_s^\infty f(z)dz + \frac{f(s)}{p}s \\
&= 1 + is
\end{aligned}$$

So,

$$\begin{aligned}
Var(Z|Z > s) &= E(Z^2|Z > s) - [E(Z|Z > s)]^2 \\
&= 1 + is - i^2 \\
&= 1 - i(i - s)
\end{aligned}$$

Results for Y

Results for Y follow from the fact that

$$\mu_Y + V_Y^{1/2}Z \sim N(\mu_Y, V_Y)$$

So, let

$$Y = \mu_Y + V_Y^{1/2}Z,$$

Then, the condition

$$Y > t$$

is equivalent to

$$\begin{aligned} \mu_Y + V_Y^{1/2}Z &> t \\ V_Y^{1/2}Z &> t - \mu_Y \\ Z &> \frac{t - \mu_Y}{V_Y^{1/2}} \\ Z &> s \end{aligned}$$

Then,

$$\begin{aligned} E(Y|Y > t) &= E(\mu_Y + V_Y^{1/2}Z|Z > s) \\ &= \mu_Y + V_Y^{1/2}i, \end{aligned}$$

and

$$\begin{aligned} Var(Y|Y > t) &= Var(\mu_Y + V_Y^{1/2}Z|Z > s) \\ &= V_Y[1 - i(i - s)] \end{aligned}$$

Numerical Example

```
In [39]: μ = 10
σ = 10
t = 15
s = (t-μ)/σ
d = Normal(0.0,1.0)
i = pdf(d,s)/(1-cdf(d,s))
meanTruncatedNormal = μ + σ*i
variTruncatedNormal = σ*σ*(1 - i*(i-s))
@printf "mean      = %8.2f  \n" meanTruncatedNormal
@printf "variance = %8.2f  \n" variTruncatedNormal

mean      =      21.41
variance =      26.85
```

Monte-Carlo Approach:

```
In [43]: using Distributions
μ = 10
σ = 10
z = rand(Normal(μ,σ),10000);

In [56]: mcmcMean = mean(z[z.>t])
mcmcVar = var(z[z.>t])
@printf "MC mean      = %8.2f  \n" mcmcMean
@printf "MC variance = %8.2f  \n" mcmcVar

MC mean      =      21.34
MC variance =      25.78
```

Bivariate Normal Example

Let $(Y) \sim N(\mu, \mathbf{V})$

$$\mu = \begin{bmatrix} 10 \\ 20 \end{bmatrix}, \mathbf{V} = \begin{bmatrix} 100 & 50 \\ 50 & 200 \end{bmatrix}$$

```
In [54]: μ = [10.0;20.0]
v = [100.0 50.0
      50.0 200.0]
d = MvNormal(μ,v)
XY = rand(d,10000)'

Out[54]: 10000x2 Array{Float64,2}:
 10.3117    41.2371
 8.49604   30.121
 1.49591    5.04669
 2.0137     21.2858
 8.12043   9.99512
 17.9018   16.9568
 1.01726   20.0321
 -8.29162   40.2454
 14.6496   45.1535
 13.9381   12.9118
 -0.612875  24.1609
 20.5875   15.1366
 16.2409   25.9275
 :
 3.98896   3.67185
 13.8927   24.0219
 3.93784   11.8521
 3.83364   4.41762
 20.7947   37.1139
 9.11036   15.7678
 4.45919   32.2166
 19.5114   21.9018
 12.777    29.3537
 18.1348   11.6092
 0.640994  14.6436
 3.39195   27.4398
```

```
In [111]: sel = XY[:,1]>10
          xxy= [XY sel]

Out[111]: 10000x3 Array{Float64,2}:
 10.3117    41.2371    1.0
  8.49604   30.121     0.0
  1.49591   5.04669   0.0
  2.0137    21.2858   0.0
  8.12043   9.99512   0.0
 17.9018    16.9568   1.0
  1.01726   20.0321   0.0
 -8.29162   40.2454   0.0
 14.6496    45.1535   1.0
 13.9381    12.9118   1.0
 -0.612875  24.1609   0.0
 20.5875    15.1366   1.0
 16.2409    25.9275   1.0
 :
  3.98896   3.67185   0.0
 13.8927    24.0219   1.0
  3.93784   11.8521   0.0
  3.83364   4.41762   0.0
 20.7947    37.1139   1.0
  9.11036   15.7678   0.0
  4.45919   32.2166   0.0
 19.5114    21.9018   1.0
 12.777     29.3537   1.0
 18.1348    11.6092   1.0
  0.640994  14.6436   0.0
  3.39195   27.4398   0.0
```

```
In [115]: (xxy[:,1][xxy[:,3]==1])
```

```
Out[115]: 18.03854352069298
```

```
In [59]: selY = XY[sel,2]

Out[59]: 5026-element Array{Float64,1}:
 41.2371
 16.9568
 45.1535
 12.9118
 15.1366
 25.9275
 17.4284
 20.6601
 44.2587
 7.21451
 26.9525
 29.502
 41.1791
  :
 41.4734
 20.1128
 33.6962
 17.7152
 16.6372
 48.6728
 27.0785
 24.0219
 37.1139
 21.9018
 29.3537
 11.6092
```

```
In [60]: mean(selY[selY.>30])
```

```
Out[60]: 38.95540792778809
```

```
In [61]: var(selY[selY.>30])
```

```
Out[61]: 52.61527300087836
```

Markov Chain Monte-Carlo Methods

- Often no closed form for $f(\theta|y)$
- Further, even if computing $f(\theta|y)$ is feasible, obtaining $f(\theta_i|y)$ would require integrating over many dimensions
- Thus, in many situations, inferences are made using the empirical posterior constructed by drawing samples from $f(\theta|y)$
- Gibbs sampler is widely used for drawing samples from posteriors

Gibbs Sampler

- Want to draw samples from $f(x_1, x_2, \dots, x_n)$
- Even though it may be possible to compute $f(x_1, x_2, \dots, x_n)$, it is difficult to draw samples directly from $f(x_1, x_2, \dots, x_n)$
- Gibbs:
 - Get valid a starting point \mathbf{x}^0
 - Draw sample \mathbf{x}^t as:

$$\begin{aligned}x_1^t &\text{ from } f(x_1 | x_2^{t-1}, x_3^{t-1}, \dots, x_n^{t-1}) \\x_2^t &\text{ from } f(x_2 | x_1^t, x_3^{t-1}, \dots, x_n^{t-1}) \\x_3^t &\text{ from } f(x_3 | x_1^t, x_2^t, \dots, x_n^{t-1}) \\&\vdots && \vdots \\x_n^t &\text{ from } f(x_n | x_1^t, x_2^t, \dots, x_{n-1}^t)\end{aligned}$$
- The sequence $\mathbf{x}^1, \mathbf{x}^2, \dots, \mathbf{x}^n$ is a Markov chain with stationary distribution $f(x_1, x_2, \dots, x_n)$

Making Inferences from Markov Chain

Can show that samples obtained from a Markov chain can be used to draw inferences from $f(x_1, x_2, \dots, x_n)$ provided the chain is:

- Irreducible: can move from any state i to any other state j
- Positive recurrent: return time to any state has finite expectation
- Markov Chains*, J. R. Norris (1997)

Bivariate Normal Example

Let $f(\mathbf{x})$ be a bivariate normal density with means

$$\mu' = [1 \quad 2]$$

and covariance matrix

$$\mathbf{V} = \begin{bmatrix} 1 & 0.5 \\ 0.5 & 2.0 \end{bmatrix}$$

Suppose we do not know how to draw samples from $f(\mathbf{x})$, but know how to draw samples from $f(x_i | x_j)$, which is univariate normal with mean:

$$\mu_{ij} = \mu_i + \frac{\nu_{ij}}{\nu_{jj}}(x_j - \mu_j)$$

and variance

$$\nu_{ij} = \nu_{ii} - \frac{\nu_{ij}^2}{\nu_{jj}}$$

```
In [125]: m = fill(0,2)
nSamples = 2000
m = [1.0, 2.0]
v = [1.0 0.5; 0.5 2.0]
y = fill(0.0,2)
sum = fill(0.0,2)
s12 = sqrt(v[1,1] - v[1,2]*v[1,2]/v[2,2])
s21 = sqrt(v[2,2] - v[1,2]*v[1,2]/v[1,1])
m1 = 0
m2 = 0;
for (iter in 1:nSamples)
    m12 = m[1] + v[1,2]/v[2,2]*(y[2] - m[2])
    m21 = m[2] + v[1,2]/v[1,1]*(y[1] - m[1])
    y[1] = rand(Normal(m12,s12),1)[1]
    y[2] = rand(Normal(m21,s21),1)[1]
    sum += y
    mean = sum/iter
    if iter%100 == 0
        @printf "%10d %8.2f %8.2f \n" iter mean[1] mean[2]
    end
end

100      1.09      2.21
200      1.06      2.16
300      1.06      2.16
400      1.05      2.12
500      1.03      2.11
600      1.01      2.10
700      1.00      2.09
800      1.01      2.09
900      1.00      2.08
1000     1.02      2.10
1100     1.00      2.09
1200     1.01      2.08
1300     1.01      2.08
1400     1.02      2.08
1500     1.03      2.10
1600     1.02      2.08
1700     1.02      2.08
1800     1.02      2.08
1900     1.03      2.07
2000     1.02      2.06
```

Metropolis-Hastings Algorithm

- Sometimes may not be able to draw samples directly from $f(x_i | \mathbf{x}_{i-1})$
- Convergence of the Gibbs sampler may be too slow
- Metropolis-Hastings (MH) for sampling from $f(\mathbf{x})$:
 - a candidate sample, y , is drawn from a proposal distribution $q(y|x^{t-1})$

$$x^t = \begin{cases} y & \text{with probability } \alpha \\ x^{t-1} & \text{with probability } 1 - \alpha \end{cases}$$

$$\alpha = \min\left(1, \frac{f(y)q(x^{t-1}|y)}{f(x^{t-1})q(y|x^{t-1})}\right)$$

- The samples from MH is a Markov chain with stationary distribution $f(x)$

Bivariate Normal Example

```
In [127]: nSamples = 10000
m = [1.0, 2.0]
v = [1.0 0.5; 0.5 2.0]
vi = inv(v)
y = fill(0.0,2)
sum = fill(0.0,2)

m1 = 0
m2 = 0
xx = 0
yl = 0
delta = 1.0
min1 = -delta*sqrt(v[1,1])
max1 = +delta*sqrt(v[1,1])
min2 = -delta*sqrt(v[2,2])
max2 = +delta*sqrt(v[2,2])
z = y-m
denOld = exp(-0.5*z'*vi*z)
d1 = Uniform(min1,max1)
d2 = Uniform(min2,max2)
ynew = fill(0.0,2);
for (iter in 1:nSamples)
    ynew[1] = y[1] + rand(d1,1)[1]
    ynew[2] = y[2] + rand(d2,1)[1]
    denNew = exp(-0.5*(ynew-m)'*vi*(ynew-m));
    alpha = denNew/denOld;
    u = rand()
    if (u < alpha[1])
        y = copy(ynew)
        denOld = exp(-0.5*(y-m)'*vi*(y-m))
    end
    sum += y
    mean = sum/iter
    if iter%1000 == 0
        @printf "%10d %8.2f %8.2f \n" iter mean[1] mean[2]
    end
end
```

1000	1.04	1.93
2000	1.10	1.91
3000	1.13	1.91
4000	1.13	1.98
5000	1.05	1.96
6000	1.03	1.94
7000	1.03	1.96
8000	1.03	1.96
9000	1.04	1.96
10000	1.06	1.97

Pedigree Package

Rohan L. Fernando

May 2015

Install PedModule

Do this only once

```
In [1]: Pkg.clone("https://github.com/reworkhow/PedModule.jl.git")
INFO: Cloning PedModule from https://github.com/reworkhow/PedModule.jl.git
INFO: Computing changes...
```

```
In [2]: using PedModule
```

```
In [3]: ;cat pedFile
```

```
1 0 0
2 0 0
3 0 0
4 1 2
5 1 2
6 1 3
```

```
In [4]: ped = PedModule.mkPed("pedFile")
ped.idMap
```

```
Out[4]: Dict{Any,Any} with 6 entries:
"4" => PedNode(3,"1","2",0.0)
"1" => PedNode(1,"0","0",0.0)
"5" => PedNode(4,"1","2",0.0)
"2" => PedNode(2,"0","0",0.0)
"6" => PedNode(6,"1","3",0.0)
"3" => PedNode(5,"0","0",0.0)
```

```
In [5]: Ai = PedModule.AInverse(ped)
```

```
Out[5]: 6x6 sparse matrix with 22 Float64 entries:
```

```
[1, 1] = 2.5  
[2, 1] = 1.0  
[3, 1] = -1.0  
[4, 1] = -1.0  
[5, 1] = 0.5  
[6, 1] = -1.0  
[1, 2] = 1.0  
[2, 2] = 2.0  
[3, 2] = -1.0  
[4, 2] = -1.0  
:  
[2, 3] = -1.0  
[3, 3] = 2.0  
[1, 4] = -1.0  
[2, 4] = -1.0  
[4, 4] = 2.0  
[1, 5] = 0.5  
[5, 5] = 1.5  
[6, 5] = -1.0  
[1, 6] = -1.0  
[5, 6] = -1.0  
[6, 6] = 2.0
```

```
In [6]: full(Ai)
```

```
Out[6]: 6x6 Array{Float64,2}:
```

```
2.5 1.0 -1.0 -1.0 0.5 -1.0  
1.0 2.0 -1.0 -1.0 0.0 0.0  
-1.0 -1.0 2.0 0.0 0.0 0.0  
-1.0 -1.0 0.0 2.0 0.0 0.0  
0.5 0.0 0.0 0.0 1.5 -1.0  
-1.0 0.0 0.0 0.0 -1.0 2.0
```

```
In [7]: A = round(inv(full(Ai)),2)
```

```
Out[7]: 6x6 Array{Float64,2}:
```

```
1.0 0.0 0.5 0.5 0.0 0.5  
0.0 1.0 0.5 0.5 -0.0 -0.0  
0.5 0.5 1.0 0.5 0.0 0.25  
0.5 0.5 0.5 1.0 0.0 0.25  
0.0 0.0 0.0 0.0 1.0 0.5  
0.5 0.0 0.25 0.25 0.5 1.0
```

Pedigree-based mixed linear models

The Prediction Problem

Model Equation

$$\mathbf{y} = \mathbf{X}\mathbf{b} + \mathbf{Z}\mathbf{u} + \mathbf{e}$$

Other aspects of the model

First moments $E[\mathbf{u}] = \mathbf{0}, E[\mathbf{e}] = \mathbf{0}$, therefore $E[\mathbf{y}] = \mathbf{X}\mathbf{b}$

Second moments $\text{var}[\mathbf{u}] = \mathbf{G}, \text{var}[\mathbf{e}] = \mathbf{R}, \text{cov}[\mathbf{u}, \mathbf{e}] = \mathbf{0}$

Distributional Assumptions e.g. $\mathbf{u}, \mathbf{e} \sim \text{MVN}$

Want to predict \mathbf{u} or linear functions like $\mathbf{k}'\mathbf{u}$

Original Solution

Generalized Least Squares (GLS)

For estimable $\mathbf{q}'\mathbf{b}$, $\mathbf{q}'\hat{\mathbf{b}}^0$ is BLUE (Best Linear Unbiased Estimator)
where $\hat{\mathbf{b}}^0 = (\mathbf{X}'\mathbf{V}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{V}'\mathbf{y}$ for $\mathbf{V} = \mathbf{Z}\mathbf{G}\mathbf{Z}' + \mathbf{R}$

then $\hat{\mathbf{u}} = \mathbf{G}\mathbf{Z}'\mathbf{V}'^{-1}(\mathbf{y} - \mathbf{X}\hat{\mathbf{b}}^0)$. is BLUP (BLU Predictor)

(same as Selection Index/BLP except $(\mathbf{y} - \mathbf{X}\hat{\mathbf{b}}^0)$ in place of $(\mathbf{y} - \mathbf{X}\mathbf{b})$
obtained by exploiting (genetic) covariances between animals
In traditional animal breeding practice

\mathbf{G} is large and dense and determined by \mathbf{A} the numerator relp matrix
 \mathbf{V} is too big to compute $\mathbf{X}'\mathbf{V}'^{-1}$

BLP vs GLS BLUP

$$\mathbf{y} = \mathbf{X}\beta + \mathbf{Z}\mathbf{u} + \mathbf{e}$$

$\mathbf{y} - \mathbf{X}\beta = \mathbf{Z}\mathbf{u} + \mathbf{e}$, a fully random model

Selection Index Equations $\mathbf{P}\mathbf{b} = \mathbf{G}\mathbf{v}$

$\mathbf{b} = \mathbf{P}^{-1}\mathbf{G}\mathbf{v}$, defines the best linear function to predict \mathbf{u}
the "weights" are the same for every animal with the same
sources of information (ie same traits observed)

$$\text{BLP } \hat{\mathbf{u}} = \mathbf{b}'(\mathbf{y} - \mathbf{X}\beta) = \mathbf{v}\mathbf{G}\mathbf{P}^{-1}(\mathbf{y} - \mathbf{X}\beta)$$

$$\text{cf GLS BLUP } \hat{\mathbf{u}} = \mathbf{G}\mathbf{Z}'\mathbf{V}'^{-1}(\mathbf{y} - \mathbf{X}\hat{\mathbf{b}}^0)$$

Henderson's Contributions One

Developed methods to compute \mathbf{G} and \mathbf{R} from field data
Henderson's Method I (not his!), II and III
Including circumstances that involved selection

Henderson's Contributions Two

Invented the Mixed Model Equations

$$\begin{bmatrix} \mathbf{X}'\mathbf{R}^{-1}\mathbf{X} & \mathbf{X}'\mathbf{R}^{-1}\mathbf{Z} \\ \mathbf{Z}'\mathbf{R}^{-1}\mathbf{X} & \mathbf{Z}'\mathbf{R}^{-1}\mathbf{Z} + \mathbf{G}^{-1} \end{bmatrix} \begin{bmatrix} \hat{\mathbf{b}}^0 \\ \hat{\mathbf{u}} \end{bmatrix} = \begin{bmatrix} \mathbf{X}'\mathbf{R}^{-1}\mathbf{y} \\ \mathbf{Z}'\mathbf{R}^{-1}\mathbf{y} \end{bmatrix}, \text{ for full rank } \mathbf{G}$$

and jointly showed $\mathbf{k}'\hat{\mathbf{b}}^0$ and $\hat{\mathbf{u}}$ were BLUE and BLUP

Computationally tractable if \mathbf{G} and \mathbf{R} assumed diagonal or block-diagonal
(eg sire model with relationships ignored)

(Order 40 matrix takes weeks to invert by hand)

MMF typically sparse in national animal evaluation

Example NRM or A matrix

Sire1	Dam1	Sire1	Dam1	Sire2	?	Sire2	?
Offspring ₁	1	1/2	0	0	0	0	0
Offspring ₂	1/2	1	0	0	0	0	0
Offspring ₃	0	0	1	1/4	0	0	0
Offspring ₄	0	0	1/4	1	0	0	0

Sires and dams unrelated and non-inbred
Simple calculation of A_p^{-1} requires including all ancestors
and would result in a matrix of order 7 not 4

A⁻¹ matrix (animal model)

Sire1	Dam1	Sire2	?				
Sire ₁	2	1	0	-1	-1	0	0
Dam ₁	1	2	0	-1	-1	0	0
Sire ₂	0	0	1.667	0	0	-0.667	-0.667
Off ₁	-1	-1	0	2	0	0	0
Off ₂	0	-1	0	0	2	0	0
Off ₃	0	0	-0.667	0	0	1.333	0
Off ₄	0	0	-0.667	0	0	0	1.333

Ancestors w/out records are fitted for simple A⁻¹ structure

Henderson's Contributions Three

Invented an algorithm to directly form A^{-1} from a pedigree list
Then G^{-1} can be formed as a scalar product or kronecker product
define d to be "mendelian" sampling variance

$$d = (1, 3/4, 1/2) \text{ for } 0, 1 \text{ or } 2 \text{ parents known}$$

define $s^* = (-1/2, -1/2, 1)$ to represent sire (if known), dam (if known)
and individual equations
accumulate $s^* d s^*$ in the sire, dam and individual rows/columns
for every trio of animals in the pedigree list

Consequence of A⁻¹ structure

Accumulate for each animal

sire	dam	i
$\begin{bmatrix} 0.25 & 0.25 & -0.5 \\ 0.25 & 0.25 & -0.5 \\ -0.5 & -0.5 & 1 \end{bmatrix} d^{-1}$		

When both parents are known
Nonparents (ie terminal offspring)
Own equation (ie row) has 2 on diagonal, -1 in sire column -1 in dam column
Parent with one offspring
Own equation has 2+1/2 on diagonal, -1 in sire and dam columns
in addition to -1/2 in the column of its mate, -1 in column of offspring
Parent with many offspring to different mates
accumulates a large diagonal element, many small negative offdiagonals

Consider rearranging the MME

In general,

$$[\mathbf{Z}'\mathbf{R}^{-1}\mathbf{X} \quad \mathbf{Z}'\mathbf{R}^{-1}\mathbf{Z} + \mathbf{G}^{-1}] \begin{bmatrix} \hat{\mathbf{b}}^* \\ \hat{\mathbf{u}} \end{bmatrix} = [\mathbf{Z}'\mathbf{R}^{-1}\mathbf{y}]$$

or equivalently $[\mathbf{Z}'\mathbf{R}^{-1}\mathbf{Z} + \mathbf{G}^{-1}] \hat{\mathbf{u}} = [\mathbf{Z}'\mathbf{R}^{-1}(\mathbf{y} - \mathbf{X}\hat{\mathbf{b}}^*)]$

Single trait animal model $\mathbf{R} = \mathbf{I}\sigma_e^2$, $\mathbf{G} = \mathbf{A}\sigma_e^2$, $\mathbf{G}^{-1} = \mathbf{A}^{-1}\sigma_e^{-2}$

or multiplying σ_e^2 , $[\mathbf{Z}'\mathbf{Z} + \lambda\mathbf{A}^{-1}] \hat{\mathbf{u}} = [\mathbf{Z}'(\mathbf{y} - \mathbf{X}\hat{\mathbf{b}}^*)]$, with $\lambda = \frac{\sigma_e^2}{\sigma_i^2}$

Consider the MME for a nonparent

$$[\mathbf{Z}'\mathbf{Z} + \lambda\mathbf{A}^{-1}] \hat{\mathbf{u}} = [\mathbf{Z}'(\mathbf{y} - \mathbf{X}\hat{\mathbf{b}}^*)]$$

Nonparent animal with one record

$$(1+2\lambda)\hat{u}_{\text{nonpar}} - \lambda\hat{u}_{\text{sire}} - \lambda\hat{u}_{\text{dam}} = \text{adjusted}_w \mathbf{y}$$

$$\hat{u}_{\text{nonpar}} = \frac{2\lambda(\hat{u}_{\text{sire}} + \hat{u}_{\text{dam}})}{(1+2\lambda)2} + \frac{(\text{adjusted}_w \mathbf{y})}{(1+2\lambda)}$$

$$= (1-w)PA + w(\text{adjusted}_w \mathbf{y}) \text{ for } w = \frac{1}{(1+2\lambda)}$$

Consider the MME for a nonparent

$$\hat{u}_{\text{nonpar}} = (1-w)PA + w(\text{adjusted}_-y) \text{ for } w = \frac{1}{(1+2\lambda)}$$

$\lambda = \frac{1-h^2}{h^2}$ so for $h^2 = 1$, $\lambda = 0, w = 1$, (no shrinkage)

for $h^2 = \text{low}$, $\lambda = \text{big}$, $w = \text{small}$, (shrink the deviation)

Two sources of BV information are pooled

The parent average PA

The individual prediction (shrunk deviation)
with heritability influencing shrinkage

Consider the MME for a nonparent

$$[\mathbf{Z}'\mathbf{Z} + \lambda\mathbf{A}^{-1}][\hat{\mathbf{u}}] = [\mathbf{Z}'(\mathbf{y} - \mathbf{X}\hat{\mathbf{b}}^0)]$$

Nonparent animal with one record

$$\hat{u}_{\text{animal}} = (1-w)PA + w(\text{adjusted}_-y)$$

Nonparent animal with no record

$$2\lambda\hat{u}_{\text{animal}} - \lambda\hat{u}_{\text{src}} - \lambda\hat{u}_{\text{dam}} = 0$$

$$\hat{u}_{\text{animal}} = \frac{\lambda(\hat{u}_{\text{src}} + \hat{u}_{\text{dam}})}{\lambda 2} = \frac{(\hat{u}_{\text{src}} + \hat{u}_{\text{dam}})}{2} = PA$$

Reliability of nonparents

Property of BLP/BLUP is $\text{cov}(u, \hat{u}) = \text{var}(\hat{u})$ so $r^2 = \frac{\text{var}(\hat{u})}{\text{var}(u)}$

but $\hat{u}_{\text{nonparent}} = \frac{\hat{u}_{\text{src}}}{2} + \frac{\hat{u}_{\text{dam}}}{2}$, for nonparent without a record

$$\text{so } r_{\text{nonparent}}^2 = \frac{r_{\text{src}}^2}{4} + \frac{r_{\text{dam}}^2}{4} \leq \frac{1}{2}$$

Finally $\Delta G = \frac{\hat{u}_{\text{nonparent}}\sigma_z}{L}$, limiting selection response

when candidates at puberty lack phenotypic information

An option to do better

Solution

- We need a different representation of the covariance between relatives, that allows relatives other than parents to directly contribute to the prediction of nonparents without records
- The NRM or A-matrix is an expectation of relationships in the context of repeated sampling of the pedigree (conditional on pedigree)

A-matrix

- Relationship with self is 1+F (noninbred F=0)
- (Additive) relationship of ½ between non-inbred full-sibs and between parents and non-inbred offspring
- Relationship of ¼ between non-inbred half-sibs and between grandparents and offspring
- But particular individuals can have greater or lesser values
 - If we know their genotype we can compute relationships conditional on the chromosome regions they inherited

Relationship matrix					
A matrix					
$\begin{bmatrix} 1 & 0 & 5 & 5 & 5 \\ 0 & 1 & 5 & 5 & 5 \\ 5 & 5 & 1 & 5 & 5 \\ 5 & 5 & 5 & 1 & 5 \\ 5 & 5 & 5 & 5 & 1 \\ 5 & 5 & 5 & 5 & 1 \end{bmatrix}$					
Consider a sire, dam and 4 full sibs					
A-inverse matrix					
$\begin{bmatrix} 3 & 2 & -1 & -1 & -1 & -1 \\ 2 & 3 & -1 & -1 & -1 & -1 \\ -1 & -1 & 2 & 0 & 0 & 0 \\ -1 & -1 & 0 & 2 & 0 & 0 \\ -1 & -1 & 0 & 0 & 2 & 0 \\ -1 & -1 & 0 & 0 & 0 & 2 \end{bmatrix}$					

Relationship matrix					
A matrix					
$\begin{bmatrix} 1 & 0 & 5 & 5 & 5 & 5 \\ 0 & 1 & 5 & 5 & 5 & 5 \\ 5 & 5 & 1 & 5 & 5 & 5 \\ 5 & 5 & 5 & 1 & 5 & 5 \\ 5 & 5 & 5 & 5 & 1 & 5 \\ 5 & 5 & 5 & 5 & 5 & 1 \end{bmatrix}$					
G matrix					
$\begin{bmatrix} 1 & 0 & .5 & .5 & .5 & .5 \\ 0 & 1 & .5 & .5 & .5 & .5 \\ .5 & .5 & 1 & .6 & .4 & .4 \\ .5 & .5 & .6 & 1 & .4 & .4 \\ .5 & .5 & .4 & .4 & 1 & .6 \\ .5 & .5 & .4 & .4 & .6 & 1 \end{bmatrix}$					
G-inverse matrix					
$\begin{bmatrix} 3.5 & 2.5 & -1.25 & -1.25 & -1.25 & -1.25 \\ 2.5 & 3.5 & -1.25 & -1.25 & -1.25 & -1.25 \\ -1.25 & -1.25 & 2.1875 & -0.3125 & 0.3125 & 0.3125 \\ -0.3125 & 2.1875 & 0.3125 & 0.3125 & 0.3125 & 0.3125 \\ 0.3125 & 0.3125 & 2.1875 & 0.3125 & -0.3125 & -0.3125 \\ -0.3125 & 2.1875 & 0.3125 & -0.3125 & 0.3125 & 2.1875 \end{bmatrix}$					

Predict the last animal with no data					
$\begin{bmatrix} -1.25\hat{u}_{sire} & -1.25\hat{u}_{dam} & .3125\hat{u}_{sib_1} & .3125\hat{u}_{sib_2} & -.3125\hat{u}_{sib_3} & 2.1875\hat{u}_{sib_4} \end{bmatrix} = [0]$					
$\hat{u}_{individual} = \frac{1.25(\hat{u}_{sire} + \hat{u}_{dam}) - 0.3125(\hat{u}_{sib_1} + \hat{u}_{sib_2}) + 0.3125\hat{u}_{sib_3}}{2.1875}$					
<p>But to form G, we needed to know which loci/QTL contribute to variation in performance</p>					

Some MME Results					
$\begin{bmatrix} X'X & X'Z \\ Z'X & Z'Z + \lambda G^{-1} \end{bmatrix} = \begin{bmatrix} C^{11} & C^{12} \\ C^{21} & C^{22} \end{bmatrix}$					
$\text{var}(g) = G - \text{var}(\hat{g}) = G - C^{22} \quad \text{var}(\hat{g} - g) = C^{22} - r_{\hat{g}}^2 = \frac{\text{var}(\hat{g})}{\text{var}(g)}$					
$\text{var}(k'g) = k'Gk \quad \text{var}(k'\hat{g}) = k'(G - C^{22})k$					

BayesGWAS

May 12, 2015

1 Bayesian Regression Models for Whole-Genome Analyses

Meuwissen et al. (2001) introduced three regression models for whole-genome prediction of breeding value of the form

$$y_i = \mu + \sum_{j=1}^k X_{ij}\alpha_j + e_i,$$

where y_i is the phenotypic value, μ is the intercept, X_{ij} is j^{th} marker covariate of animal i , α_j is the partial regression coefficient of X_{ij} , and e_i are identically and independently distributed residuals with mean zero and variance σ_e^2 . In most current analyses, X_{ij} are SNP genotype covariates that can be coded as 0, 1 and 2, depending on the number of B alleles at SNP locus j .

In all three of their models, a flat prior was used for the intercept and a scaled inverted chi-square distribution for σ_e^2 . The three models introduced by Meuwissen et al. @Meuwissen.THE.ca.2001a differ only in the prior used for α_j .

1.1 BLUP

In their first model, which they called BLUP, a normal distribution with mean zero and known variance, σ_α^2 , is used as the prior for α_j .

1.1.1 The meaning of σ_α^2

Assume the QTL are in the marker panel. Then, the genotypic value g_i for a randomly sampled animal i can be written as

$$g_i = \mu + \mathbf{x}'_i \boldsymbol{\alpha},$$

where \mathbf{x}'_i is the vector of SNP genotype covariates and $\boldsymbol{\alpha}$ is the vector of regression coefficients. Note that randomly sampled animals differ only in \mathbf{x}'_i and have $\boldsymbol{\alpha}$ in common. Thus, genotypic variability is entirely due to variability in the genotypes of animals. So, σ_α^2 is not the genetic variance at a locus (Fernando:2007, Gianola:2009:Genetics:19620397).

1.1.2 Relationship of σ_α^2 to genetic variance

Assume loci with effect on trait are in linkage equilibrium. Then, the additive genetic variance is

$$V_A = \sum_j^k 2p_j q_j \alpha_j^2,$$

where $p_j = 1 - q_j$ is gene frequency at SNP locus j . Letting $U_j = 2p_j q_j$ and $V_j = \alpha_j^2$,

$$V_A = \sum_j^k U_j V_j.$$

For a randomly sampled locus, covariance between U_j and V_j is

$$C_{UV} = \frac{\sum_j U_j V_j}{k} - \left(\frac{\sum_j U_j}{k}\right)\left(\frac{\sum_j V_j}{k}\right)$$

Rearranging this expression for C_{UV} gives

$$\sum_j U_j V_j = kC_{UV} + \left(\sum_j U_j\right)\left(\frac{\sum_j V_j}{k}\right)$$

So,

$$V_A = kC_{UV} + \left(\sum_j 2p_j q_j\right)\left(\frac{\sum_j \alpha_j^2}{k}\right).$$

Letting $\sigma_\alpha^2 = \frac{\sum_j \alpha_j^2}{k}$ gives

$$V_A = kC_{UV} + \left(\sum_j 2p_j q_j\right)\sigma_\alpha^2$$

and

$$\sigma_\alpha^2 = \frac{V_A - kC_{UV}}{\sum_j 2p_j q_j},$$

which gives

$$\sigma_\alpha^2 = \frac{V_A}{\sum_j 2p_j q_j},$$

if gene frequency is independent of the effect of the gene.

1.1.3 Full-conditionals:

The joint posterior for all the parameters is proportional to

$$\begin{aligned} f(\theta | \mathbf{y}) &\propto f(\mathbf{y} | \theta) f(\theta) \\ &\propto (\sigma_e^2)^{-n/2} \exp \left\{ -\frac{(\mathbf{y} - \mathbf{1}\mu - \sum \mathbf{X}_j \alpha_j)'(\mathbf{y} - \mathbf{1}\mu - \sum \mathbf{X}_j \alpha_j)}{2\sigma_e^2} \right\} \\ &\times \prod_{j=1}^k (\sigma_\alpha^2)^{-1/2} \exp \left\{ -\frac{\alpha_j^2}{2\sigma_\alpha^2} \right\} \\ &\times (\sigma_\alpha^2)^{-(\nu_\alpha + 2)/2} \exp \left\{ -\frac{\nu_\alpha S_\alpha^2}{2\sigma_\alpha^2} \right\} \\ &\times (\sigma_e^2)^{-(2+\nu_e)/2} \exp \left\{ -\frac{\nu_e S_e^2}{2\sigma_e^2} \right\}, \end{aligned}$$

where θ denotes all the unknowns.

1.1.4 Full-conditional for μ

The full-conditional for μ is a normal distribution with mean $\hat{\mu}$ and variance $\frac{\sigma_e^2}{n}$, where $\hat{\mu}$ is the least-squares estimate of μ in the model

$$\mathbf{y} - \sum_{j=1}^k \mathbf{X}_j \alpha_j = \mathbf{1}\mu + \mathbf{e},$$

and $\frac{\sigma_e^2}{n}$ is the variance of this estimator (n is the number of observations).

1.1.5 Full-conditional for α_j

$$\begin{aligned}
f(\alpha_j | \text{ELSE}) &\propto \exp \left\{ -\frac{(\mathbf{w}_j - \mathbf{X}_j \alpha_j)'(\mathbf{w}_j - \mathbf{X}_j \alpha_j)}{2\sigma_e^2} \right\} \\
&\times \exp \left\{ -\frac{\alpha_j^2}{2\sigma_\alpha^2} \right\} \\
&\propto \exp \left\{ -\frac{[\mathbf{w}'_j \mathbf{w}_j - 2\mathbf{w}'_j \mathbf{X}_j \alpha_j + \alpha_j^2 (\mathbf{x}'_j \mathbf{x}_j + \sigma_e^2/\sigma_\alpha^2)]}{2\sigma_e^2} \right\} \\
&\propto \exp \left\{ -\frac{(\alpha_j - \hat{\alpha}_j)^2}{\frac{2\sigma_e^2}{(\mathbf{x}'_j \mathbf{x}_j + \sigma_e^2/\sigma_\alpha^2)}} \right\},
\end{aligned}$$

where

$$\mathbf{w}_j = \mathbf{y} - \mathbf{1}\mu - \sum_{l \neq j} \mathbf{X}_l \alpha_l.$$

So, the full-conditional for α_j is a normal distribution with mean

$$\hat{\alpha}_j = \frac{\mathbf{X}'_j \mathbf{w}_j}{(\mathbf{x}'_j \mathbf{x}_j + \sigma_e^2/\sigma_\alpha^2)}$$

and variance $\frac{\sigma_e^2}{(\mathbf{x}'_j \mathbf{x}_j + \sigma_e^2/\sigma_\alpha^2)}$.

1.1.6 Full-conditional for σ_α^2

$$\begin{aligned}
f(\sigma_\alpha^2 | \text{ELSE}) &\propto \prod_{j=1}^k (\sigma_\alpha^2)^{-1/2} \exp \left\{ -\frac{\alpha_j^2}{2\sigma_\alpha^2} \right\} \\
&\times (\sigma_\alpha^2)^{-(\nu_\alpha+2)/2} \exp \left\{ -\frac{\nu_\alpha S_\alpha^2}{2\sigma_\alpha^2} \right\} \\
&\propto (\sigma_\alpha^2)^{-(k+\nu_\alpha+2)/2} \exp \left\{ -\frac{\sum_{j=1}^k \alpha_j^2 + \nu_\alpha S_\alpha^2}{2\sigma_\alpha^2} \right\},
\end{aligned}$$

and this is proportional to a scaled inverted chi-square distribution with $\tilde{\nu}_\alpha = \nu_\alpha + k$ and scale parameter $\tilde{S}_\alpha^2 = (\sum_k \alpha_j^2 + \nu_\alpha S_\alpha^2)/\tilde{\nu}_\alpha$.

1.1.7 Full-conditional for σ_e^2

$$\begin{aligned}
f(\sigma_e^2 | \text{ELSE}) &\propto (\sigma_e^2)^{-n/2} \exp \left\{ -\frac{(\mathbf{y} - \mathbf{1}\mu - \sum \mathbf{X}_j \alpha_j)'(\mathbf{y} - \mathbf{1}\mu - \sum \mathbf{X}_j \alpha_j)}{2\sigma_e^2} \right\} \\
&\times (\sigma_e^2)^{-(2+\nu_e)/2} \exp \left\{ -\frac{\nu_e S_e^2}{2\sigma_e^2} \right\} \\
&\propto (\sigma_e^2)^{-(n+2+\nu_e)/2} \exp \left\{ -\frac{(\mathbf{y} - \mathbf{1}\mu - \sum \mathbf{X}_j \alpha_j)'(\mathbf{y} - \mathbf{1}\mu - \sum \mathbf{X}_j \alpha_j) + \nu_e S_e^2}{2\sigma_e^2} \right\}.
\end{aligned}$$

which is proportional to a scaled inverted chi-square density with $\tilde{\nu}_e = n + \nu_e$ degrees of freedom and $\tilde{S}_e^2 = \frac{(\mathbf{y} - \mathbf{1}\mu - \sum \mathbf{X}_j \alpha_j)'(\mathbf{y} - \mathbf{1}\mu - \sum \mathbf{X}_j \alpha_j) + \nu_e S_e^2}{\tilde{\nu}_e}$ scale parameter.

1.2 BayesB

1.2.1 Model

The usual model for BayesB is:

$$y_i = \mu + \sum_{j=1}^k X_{ij} \alpha_j + \epsilon_i,$$

where the prior μ is flat and the prior for α_j is a mixture distribution:

$$\alpha_j = \begin{cases} 0 & \text{probability } \pi \\ \sim N(0, \sigma_j^2) & \text{probability } (1 - \pi) \end{cases},$$

where σ_j^2 has a scaled inverted chi-square prior with scale parameter S_α^2 and ν_α degrees of freedom. The residual is normally distributed with mean zero and variance σ_ϵ^2 , which has a scaled inverted chi-square prior with scale parameter S_ϵ^2 and ν_ϵ degrees of freedom. Meuwissen et al. (Meuwissen.THE.ea.2001a) gave a Metropolis-Hastings sampler to jointly sample σ_j^2 and α_j . Here, we will show how the Gibbs sampler can be used in BayesB.

In order to use the Gibbs sampler, the model is written as

$$y_i = \mu + \sum_{j=1}^k X_{ij} \beta_j \delta_j + \epsilon_i,$$

where $\beta_j \sim N(0, \sigma_j^2)$ and δ_j is Bernoulli($1 - \pi$):

$$\delta_j = \begin{cases} 0 & \text{probability } \pi \\ 1 & \text{probability } (1 - \pi) \end{cases}.$$

Other priors are the same as in the usual model. Note that in this model, $\alpha_j = \beta_j \delta_j$ has a mixture distribution as in the usual BayesB model.

1.2.2 Full-conditionals:

The joint posterior for all the parameters is proportional to

$$\begin{aligned} f(\theta | \mathbf{y}) &\propto f(\mathbf{y} | \theta) f(\theta) \\ &\propto (\sigma_\epsilon^2)^{-n/2} \exp \left\{ -\frac{(\mathbf{y} - \mathbf{1}\mu - \sum \mathbf{X}_j \beta_j \delta_j)'(\mathbf{y} - \mathbf{1}\mu - \sum \mathbf{X}_j \beta_j \delta_j)}{2\sigma_\epsilon^2} \right\} \\ &\times \prod_{j=1}^k (\sigma_j^2)^{-1/2} \exp \left\{ -\frac{\beta_j^2}{2\sigma_j^2} \right\} \\ &\times \prod_{j=1}^k \pi^{(1-\delta_j)} (1-\pi)^{\delta_j} \\ &\times \prod_{j=1}^k (\sigma_j^2)^{-(\nu_\beta+2)/2} \exp \left\{ -\frac{\nu_\beta S_\beta^2}{2\sigma_j^2} \right\} \\ &\times (\sigma_\epsilon^2)^{-(2+\nu_\epsilon)/2} \exp \left\{ -\frac{\nu_\epsilon S_\epsilon^2}{2\sigma_\epsilon^2} \right\}. \end{aligned}$$

where θ denotes all the unknowns.

1.2.3 Full-conditional for μ

The full-conditional for μ is a normal distribution with mean $\hat{\mu}$ and variance $\frac{\sigma_\epsilon^2}{n}$, where $\hat{\mu}$ is the least-squares estimate of μ in the model

$$\mathbf{y} - \sum_{j=1}^k \mathbf{X}_j \beta_j \delta_j = \mathbf{1}\mu + \mathbf{e},$$

and $\frac{\sigma_\epsilon^2}{n}$ is the variance of this estimator (n is the number of observations).

1.2.4 Full-conditional for β_j

$$\begin{aligned}
f(\beta_j | \text{ELSE}) &\propto \exp \left\{ -\frac{(\mathbf{w}_j - \mathbf{X}_j \beta_j \delta_j)' (\mathbf{w}_j - \mathbf{X}_j \beta_j \delta_j)}{2\sigma_e^2} \right\} \\
&\times \exp \left\{ -\frac{\beta_j^2}{2\sigma_j^2} \right\} \\
&\times \exp \left\{ -\frac{[\mathbf{w}'_j \mathbf{w}_j - 2\mathbf{w}'_j \mathbf{X}_j \beta_j \delta_j + \beta_j^2 (\mathbf{x}'_j \mathbf{x}_j \delta_j + \sigma_e^2 / \sigma_j^2)]}{2\sigma_e^2} \right\} \\
&\times \exp \left\{ -\frac{(\beta_j - \hat{\beta}_j)^2}{\frac{2\sigma_e^2}{(\mathbf{x}'_j \mathbf{x}_j \delta_j + \sigma_e^2 / \sigma_j^2)}} \right\}.
\end{aligned}$$

where

$$\mathbf{w}_j = \mathbf{y} - \mathbf{1}\mu - \sum_{l \neq j} \mathbf{X}_l \beta_l \delta_l.$$

So, the full-conditional for β_j is a normal distribution with mean

$$\hat{\beta}_j = \frac{\mathbf{X}'_j \mathbf{w}_j \delta_j}{(\mathbf{x}'_j \mathbf{x}_j \delta_j + \sigma_e^2 / \sigma_j^2)}$$

and variance $\frac{\sigma_e^2}{(\mathbf{x}'_j \mathbf{x}_j \delta_j + \sigma_e^2 / \sigma_j^2)}$.

1.2.5 Full-conditional for δ_j

$$\Pr(\delta_j = 1 | \text{ELSE}) \propto \frac{h(\delta_j = 1)}{h(\delta_j = 1) + h(\delta_j = 0)},$$

where

$$h(\delta_j) = \pi^{(1-\delta_j)} (1-\pi)^{\delta_j} \exp \left\{ -\frac{(\mathbf{w}_j - \mathbf{X}_j \beta_j \delta_j)' (\mathbf{w}_j - \mathbf{X}_j \beta_j \delta_j)}{2\sigma_e^2} \right\},$$

1.2.6 Full-conditional for σ_j^2

$$\begin{aligned}
f(\sigma_j^2 | \text{ELSE}) &\propto (\sigma_j^2)^{-1/2} \exp \left\{ -\frac{\beta_j^2}{2\sigma_j^2} \right\} \\
&\times (\sigma_j^2)^{-(\nu_\beta + 2)/2} \exp \left\{ -\frac{\nu_\beta S_\beta^2}{2\sigma_j^2} \right\} \\
&\propto (\sigma_j^2)^{-(1+\nu_\beta + 2)/2} \exp \left\{ -\frac{\beta_j^2 + \nu_\beta S_\beta^2}{2\sigma_j^2} \right\},
\end{aligned}$$

and this is proportional to a scaled inverted chi-square distribution with $\tilde{\nu}_j = \nu_\beta + 1$ and scale parameter $\tilde{S}_j^2 = (\beta_j^2 + \nu_\beta S_\beta^2) / \tilde{\nu}_j$.

1.2.7 Full-conditional for σ_e^2

$$\begin{aligned}
f(\sigma_e^2 | \text{ELSE}) &\propto (\sigma_e^2)^{-n/2} \exp \left\{ -\frac{(\mathbf{y} - \mathbf{1}\mu - \sum \mathbf{X}_j \beta_j \delta_j)' (\mathbf{y} - \mathbf{1}\mu - \sum \mathbf{X}_j \beta_j \delta_j)}{2\sigma_e^2} \right\} \\
&\times (\sigma_e^2)^{-(2+\nu_e)/2} \exp \left\{ -\frac{\nu_e S_e^2}{2\sigma_e^2} \right\} \\
&\propto (\sigma_e^2)^{-(n+2+\nu_e)/2} \exp \left\{ -\frac{(\mathbf{y} - \mathbf{1}\mu - \sum \mathbf{X}_j \beta_j \delta_j)' (\mathbf{y} - \mathbf{1}\mu - \sum \mathbf{X}_j \beta_j \delta_j) + \nu_e S_e^2}{2\sigma_e^2} \right\}.
\end{aligned}$$

which is proportional to a scaled inverted chi-square density with $\hat{\nu}_e = n + \nu_e$ degrees of freedom and $\hat{S}_e^2 = \frac{(\mathbf{y} - \mathbf{1}\mu - \sum \mathbf{X}_j \beta_j \delta_j)'(\mathbf{y} - \mathbf{1}\mu - \sum \mathbf{X}_j \beta_j \delta_j) + \nu_e S_e^2}{\hat{\nu}_e}$ scale parameter.

BayesC0

Simulating Genotypes and Phenotypes

```
In [31]: using(Distributions)
```

```
In [2]: nObs      = 100
nMarkers  = 1000
X = sample([0,1,2],(nObs,nMarkers))
α  = randn(nMarkers)
a  = X*α
stdGen = std(a)
a = a/stdGen
y = a + randn(nObs)
saveAlpha = α
nothing
```

Centering Genotype Covariates

```
In [3]: meanXCols = mean(X,1)
X = X - ones(nObs,1)*meanXCols;
```

Priors

```
In [4]: seed          = 10    # set the seed for the random number generator
chainLength    = 2000   # number of iterations
probFixed      = 0      # parameter "pi" the probability SNP effect is z
dfEffectVar    = 4      # hyper parameter (degrees of freedom) for locus
nuRes          = 4      # hyper parameter (degrees of freedom) for residual
varGenotypic   = 1      # used to derive hyper parameter (scale) for location
varResidual    = 1      # used to derive hyper parameter (scale) for residual
scaleVar        = varGenotypic*(dfEffectVar-2)/dfEffectVar   # scale factor for location
scaleRes        = varResidual*(nuRes-2)/nuRes                 # scale factor for residual
nothing
```

Function for Sampling Marker Effects

```
In [5]: function get_column(X,nRows,j)
    indx = 1 + (j-1)*nRows
    ptr = pointer(X,indx)
    pointer_to_array(ptr,nRows)
end

Out[5]: get_column (generic function with 1 method)
```

```
In [6]: xpx = [(X[:,i]'X[:,i])[1]::Float64 for i=1:nMarkers]
xArray = Array(Array{Float64,1},nMarkers)
for i=1:nMarkers
    xArray[i] = get_column(X,nObs,i)
end
```

```
In [7]: typeof(xArray[1])
Out[7]: Array{Float64,1}
```

Computing the adjusted right-hand-side efficiently

We want to compute:

$$rhs = \mathbf{X}_j'(\mathbf{y}_{corr} + \mathbf{X}_j\alpha_j)$$

This is more efficiently obtained as

$$rhs = \mathbf{X}_j'\mathbf{y}_{corr} + \mathbf{X}_j'\mathbf{X}_j\alpha_j,$$

using the diagonals of $\mathbf{X}'\mathbf{X}$ that have already been computed (line 4 of the function below).

```
In [19]: 1 function sampleEffects!(nMarkers,xArray,xpx,yCorr,alpha,meanAlpha,vare,var
 2     nObs = size(X,1)
 3     for j=1:nMarkers
 4         rhs::Float64 = dot(xArray[j],yCorr) + xpx[j]*alpha[j]
 5         lhs::Float64 = xpx[j] + vare/varEffects
 6         invLhs::Float64 = 1.0/lhs
 7         mean::Float64 = invLhs*rhs
 8         oldAlpha::Float64 = alpha[j]
 9         alpha[j] = mean + randn()*sqrt(invLhs*vare)
10         BLAS.axpy!(oldAlpha-alpha[j],xArray[j],yCorr)
11     end
12     nothing
13 end
```

```
Out[19]: sampleEffects! (generic function with 1 method)
```

Function for BayesC0

The intercept is sampled first and the `sampleEffects!` function is called to sample the marker effects

```
In [10]: ch1=Chisq(nObs+nuRes)
chi2=Chisq(dfEffectVar+nMarkers)

function BayesC0!(numIter,nMarkers,x,xpx,yCorr,mu,meanMu,alpha,meanAlpha,vare,
    for i=1:numIter
        # sample residual variance
        vare = (dot(yCorr,yCorr)+nuRes*scaleRes)/rand(ch1)

        # sample intercept
        yCorr = yCorr+mu
        rhs = sum(yCorr)
        invLhs = 1.0/(nObs)
        mean = rhs*invLhs
        mu = mean + randn()*sqrt(invLhs*vare)
        yCorr = yCorr - mu
        meanMu = meanMu + (mu - meanMu)/i

        # sample effects
        sampleEffects!(nMarkers,xArray,xpx,yCorr,alpha,meanAlpha,vare,varEffects)
        meanAlpha = meanAlpha + (alpha - meanAlpha)/i

        #sample locus effect variance
        varEffects = (scaleVar*dfEffectVar + dot(alpha,alpha))/rand(chi2)

        if (i%100)==0
            yhat = meanMu+X*meanAlpha
            resCorr = cor(a,yhat)
            println ("Correlation of between true and predicted breeding v
        end
    end
end
```

```
Out[10]: BayesC0! (generic function with 1 method)
```

Run BayesC0

```
In [30]: meanMu      = 0
meanAlpha = zeros(nMarkers)

#initial values
vare = 1
varEffects = 1
mu = mean(y)
yCorr = y - mu
alpha = fill(0.0,nMarkers)

#run it
@time BayesC0!(chainLength,nMarkers,X,xpx,yCorr,mu,meanMu,alpha,meanAlpha,
Correlation of between true and predicted breeding value: 0.77452987300536
Correlation of between true and predicted breeding value: 0.77472194735639
elapsed time: 0.213988087 seconds (53211392 bytes allocated, 12.66% gc time)
```

Compare Runtime with R Implementation

```
In [18]: ;Rscript RBayesC0/BayesC0.R
```

user	system	elapsed
50.936	1.524	52.569

```
In [32]: ;cat RBayesC0/BayesC0.R
```

```
# This code is for illustrative purposes and not efficient for large problems
# Real life data analysis (using the same file formats) is available at
# bigs.ansci.iastate.edu/login.html based on GenSel cpp software implementation
#
# Rohan Fernando (rohan@iastate.edu)
# Dorian Garrick (dorian@iastate.edu)
# copyright August 2012

# Parameters
setwd("RBayesC0")
seed          = 10    # set the seed for the random number generator
chainLength   = 2000  # number of iterations
dfEffectVar   = 4     # hyper parameter (degrees of freedom) for locus
nuRes         = 4     # hyper parameter (degrees of freedom) for residual
varGenotypic  = 1     # used to derive hyper parameter (scale) for location
varResidual   = 1     # used to derive hyper parameter (scale) for residuals
windowSize    = 10    # number of consecutive markers in a genomic window
outputFrequency = 100  # frequency for reporting performance and for convergence

markerFileName      = "genotypes.dat"
trainPhenotypeFileName = "trainPhenotypes.dat"
testPhenotypeFileName = "testPhenotypes.dat"
```

```

set.seed(seed)

genotypeFile      = read.table(markerFileName, header=TRUE)
trainPhenotypeFile = read.table(trainPhenotypeFileName, skip=1)[,1:2]
testPhenotypeFile = read.table(testPhenotypeFileName, skip=1)[,1:2]
commonTrainingData = merge(trainPhenotypeFile, genotypeFile, by.x=1, by.y=1)
commonTestData     = merge(testPhenotypeFile, genotypeFile, by.x=1, by.y=1)

remove(genotypeFile)                                # Free
remove(trainPhenotypeFile)                          # Free
remove(testPhenotypeFile)                           # Free
animalID = unname(as.matrix(commonTrainingData[,1]))    # First
y        = commonTrainingData[, 2]                   # Second f
Z        = commonTrainingData[, 3: ncol(commonTrainingData)]   # Remainin
Z        = unname(as.matrix((Z + 10)/10));           # Recode g
markerID = colnames(commonTrainingData)[3:ncol(commonTrainingData)] # Reme
remove(commonTrainingData)

testID = unname(as.matrix(commonTestData[,1]))          # First fi
yTest    = commonTestData[, 2]                         # Second f
ZTest    = commonTestData[, 3: ncol(commonTestData)]    # Remainin
ZTest    = unname(as.matrix((ZTest + 10)/10));         # Recode g
remove(commonTestData)

nmarkers = ncol(Z)                                     # number c
nrecords = nrow(Z)                                    # number c

# center the genotype matrix to accelerate mixing
markerMeans = colMeans(Z)                            # compute the mean f
Z = t(t(Z) - markerMeans)                          # deviate covariate
p = markerMeans/2.0                                 # compute frequency
mean2pq = mean(2*p*(1-p))                          # compute mean genot

varEffects = varGenotypic/(nmarkers*mean2pq)        # variance of locus
#(e.g. Fernando et al 192-195)
scaleVar   = varEffects*(dfEffectVar-2)/dfEffectVar; # scale factor for l
scaleRes   = varResidual*(nuRes-2)/nuRes            # scale factor for r

numberWindows = nmarkers/windowSize                 # number of genomic windows
numberSamples = chainLength/outputFrequency        # number of samples

alpha       = array(0.0, nmarkers) # reserve a vector to store sampled
meanAlpha  = array(0.0, nmarkers) # reserve a vector to accumulate th
modelFreq  = array(0.0, nmarkers) # reserve a vector to store model f

```

```

mu           = mean(y)                      # starting value for the location p
meanMu       = 0                           # reserve a scalar to accumulate th
geneticVar   = array(0,numberSamples) # reserve a vector to store sampl
                                         # reserve a matrix to store sampled
windowVarProp = matrix(0,nrow=numberSamples,ncol=numberWindows)
sampleCount  = 0                           # initialize counter for number of

# adjust y for the fixed effect (ie location parameter)
ycorr = y - mu

ZPZ=t(Z)%%Z
zpz=diag(ZPZ)

ptime=proc.time()
# mcmc sampling
for (iter in 1:chainLength){

  # sample residual variance
  vare = ( t(ycorr)%%ycorr + nuRes*scaleRes )/rchisq(1,nrecords + n

  # sample intercept
  ycorr = ycorr + mu                         # Unadjust y for the previous
  rhs   = sum(ycorr)                         # Form X'y
  invLhs = 1.0/nrecords                      # Form (X'X)-1
  mean = rhs*invLhs                          # Solve (X'X) mu = X'y
  mu = rnorm(1,mean,sqrt(invLhs*vare))      # Sample new location parameter
  ycorr = ycorr - mu                         # Adjust y for the new sample
  meanMu = meanMu + mu                       # Accumulate the sum to compute

  # sample effect for each locus
  for (locus in 1:nmarkers){

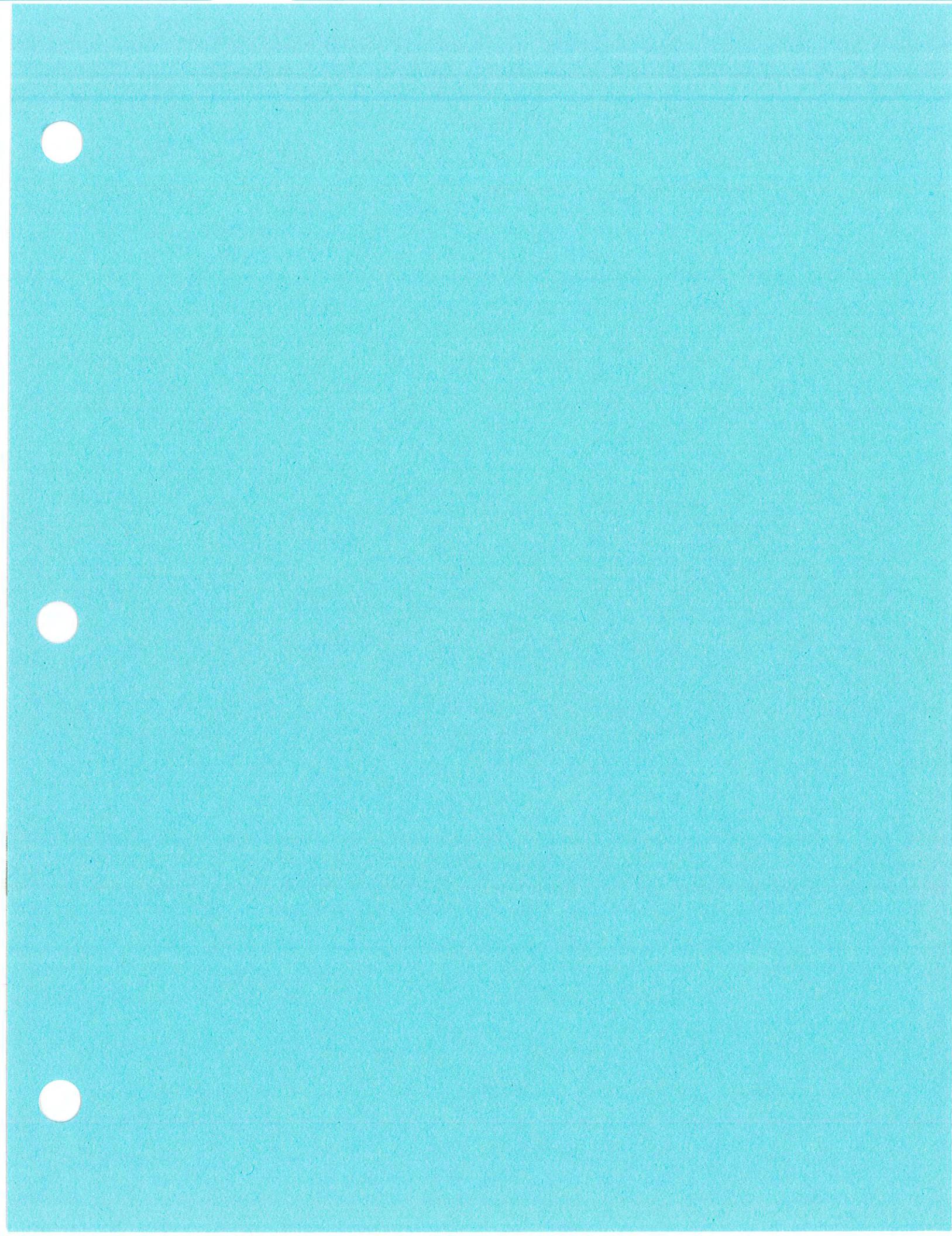
    rhs=t(Z[,locus])%%ycorr + zpz[locus]*alpha[locus]
    mmeLhs = zpz[locus] + vare/varEffects
    invLhs = 1.0/mmeLhs                      # Inverse of the matrix
    mean = invLhs*rhs                         # So we can solve for alpha
    oldAlpha=alpha[locus]
    alpha[locus]= rnorm(1,mean,sqrt(invLhs*vare)) # Sample new effect parameter
    ycorr = ycorr + Z[,locus]*(oldAlpha-alpha[locus]);
    meanAlpha[locus] = meanAlpha[locus] + alpha[locus]; # Accumulate the sum to compute
  }

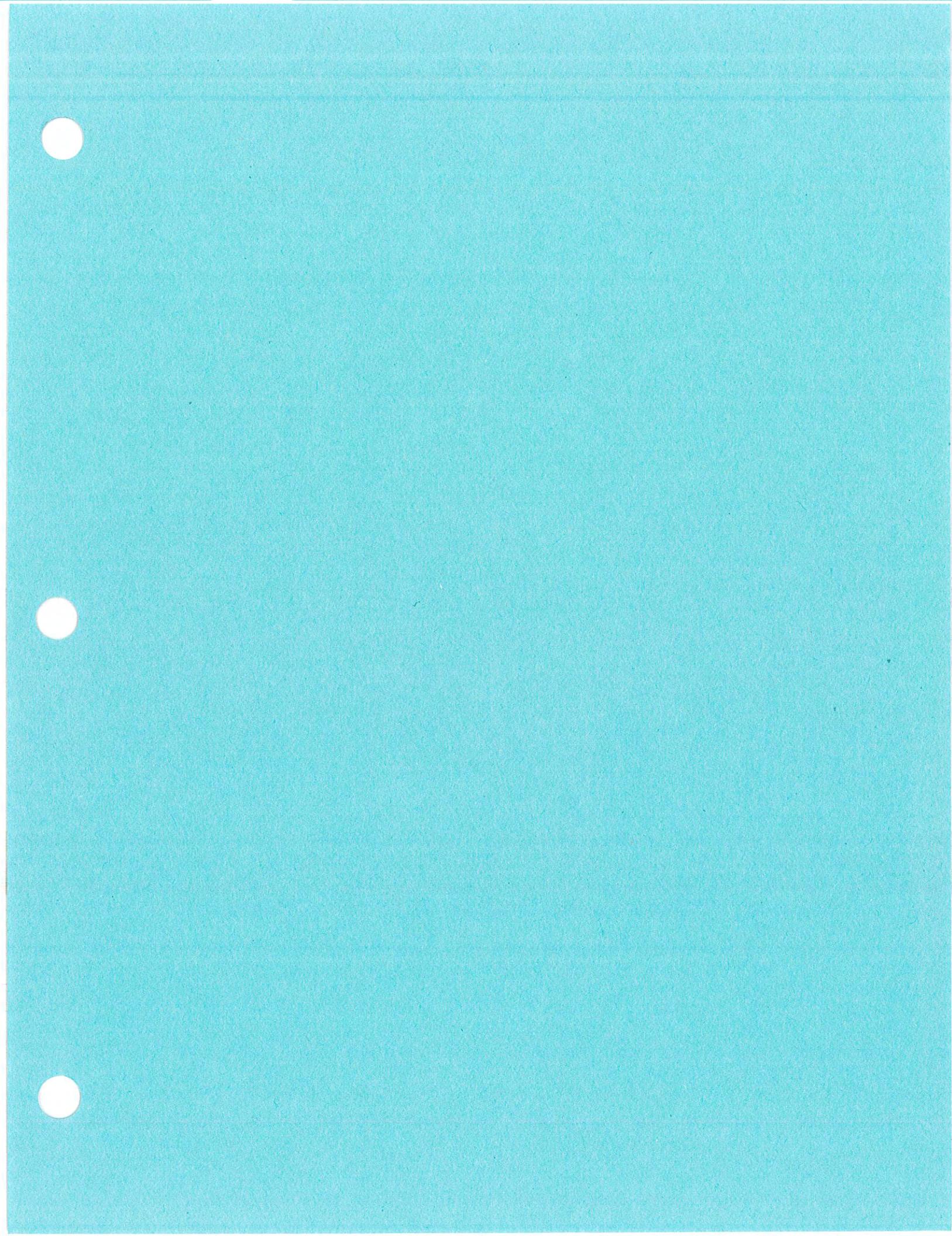
  # sample the common locus effect variance
  varEffects = ( scaleVar*dfEffectVar + sum(alpha^2) )/rchisq(1,dfEf
}

}

```

```
proc.time()-ptime
```





Application of Whole Genome Prediction Methods for Genomewide Association Studies: a Bayesian Approach

R.L. Fernando A. Toosi D.J. Garrick J.C.M. Dekkers

Department of Animal Science
Iowa State University

10th World Congress of Genetics Applied to Livestock Production

Two Approaches

- Bayesian multiple-regression models (BMR)
- Single-marker models (SM)

Compare Approaches

Model	SM	BMR
False Positives (FP)	Simple Regression	Multiple Regression
Inference	Genomewise Error Rate Frequentist	Proportion of FP Bayesian

Models

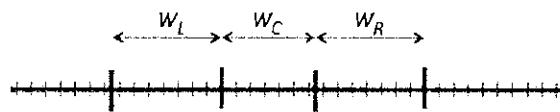
Simple Regression

- QTL may have low LD with all markers in region
- Need to explicitly model population structure

Multiple Regression

- Inference based on genomic windows
- Markers can capture population structure
 - Explicit modeling of structure results in lower power
- Inference of QTL

Composite Genomic Window



Controlling False Positives

Genomewise error rate

- Control probability of one or more false positives among all tests
- Incurs multiple-test penalty

Proportion of false positives

- Control proportion of false positives (PFP)
- Related to FDR
- No multiple-test penalty (Fernando et al., 2004; Stephens and Balding, 2009)

Definition PFP

- V number of false positives
- R number of positives
- $PFP = \frac{E(V)}{E(R)}$
- $FDR = E(\frac{V}{R} | R > 0) \Pr(R > 0)$
- If PFP is γ in each of n independent experiments, the proportion of false positives among significant results across all experiments will converge to γ as n increases.
- In general, the above property does not hold for FDR.
- PFP is a multiple test extension of the posterior type I error rate (PER).
- If PER is γ for a random test, PFP is also γ for the collection of tests.

Definition of PER

- In the frequentist approach, inference on H_0 is based on the distribution of some test statistic given H_0 is true
- posterior type I error rate (PER) is the conditional probability of H_0 being true given that, based on a statistical test, H_0 has been rejected.

$$\text{PER} = \frac{\Pr(H_0 \text{ is rejected}, H_0 \text{ is true})}{\Pr(H_0 \text{ is rejected}, H_0 \text{ is true}) + \Pr(H_0 \text{ is rejected}, H_0 \text{ is false})}$$

$$= \frac{\alpha \Pr(H_0)}{\alpha \Pr(H_0) + (1 - \beta)(1 - \Pr(H_0))}$$

α is the type I error rate, and $(1 - \beta)$ is the power of the test

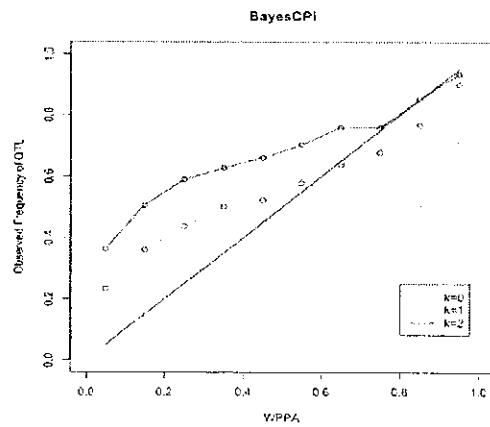
Definition of PER

- In the Bayesian approach, inference on H_0 is based on $\Pr(H_0|y)$.
- Typically, $\Pr(H_0|y)$ is estimated by counting the number of MCMC samples where H_0 is true.
- If H_0 is rejected when $\Pr(H_0|y) < \gamma$, $\text{PER} < \gamma$.
- $\Pr(H_0|y)$ is not a frequentist probability.

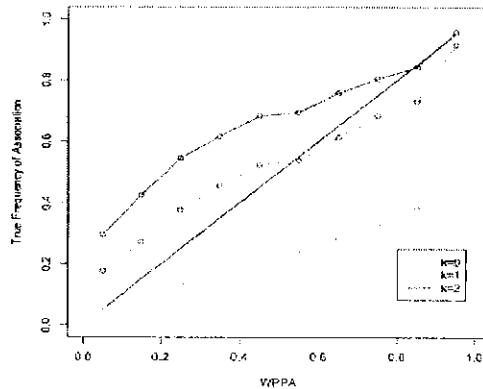
Simulation

- 52k SNP genotypes from 3,570 Angus bulls
- 100 data sets of size 1000 or 3,570 were randomly sampled
- marker effects randomly sampled according to BayesC with $\pi = 0.995$
- markers with non-zero effects (QTL) were not included in marker panel
- $h^2 = 0.9$

Results for N=1000



Results for N=3,570



Summary

- Genomic window based inference multiple regression models
- When PFP is used to manage false positives, no multiple-test penalty
- Bayesian posterior probabilities can be used to control PFP
 - $\Pr(H_0)$, and power of test can be treated as unknown
 - Do not need to know the distribution of test statistic
 - Simple to determine significance threshold

Acknowledgements

- Funding:

- NIH Grant R01GM099992
- USDA/AFRI project EBIGS

Extension to Multiple Linear Regression

Consider the multiple regression model

$$y_i = \beta_0 + \sum_j x_{ij}\beta_j + e_i \quad (2)$$

which extends model (1) to include multiple covariates x_{ij} . In matrix notation, this model can be written as
 $\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \mathbf{e}$,

where $\boldsymbol{\beta}' = [\beta_0, \beta_1, \beta_2, \dots, \beta_k]$ and the matrix \mathbf{X} contains the corresponding covariates.

Model with Normal Prior for Regression Coefficients

Here we consider a model with a flat prior for β_0 and iid normal priors for the slopes:

$$\beta_j \sim N(0, \sigma_\beta^2) \text{ for } j = 1, 2, \dots, k,$$

where σ_β^2 is assumed to be known. The residuals are assumed iid normal with null mean and variance σ_e^2 , which itself is assigned a scaled inverted chi-square prior. Then, the joint posterior for θ is

$$\begin{aligned} f(\theta | \mathbf{y}) &\propto f(\mathbf{y} | \theta) f(\theta) \\ &\propto (\sigma_e^2)^{-n/2} \exp \left\{ -\frac{(\mathbf{y} - \mathbf{X}\boldsymbol{\beta})'(\mathbf{y} - \mathbf{X}\boldsymbol{\beta})}{2\sigma_e^2} \right\} \\ &\times (\sigma_\beta^2)^{-k/2} \exp \left\{ -\frac{\sum_{j=1}^k \beta_j^2}{2\sigma_\beta^2} \right\} \\ &\times (\sigma_e^2)^{-(2+\nu_e)/2} \exp \left\{ -\frac{\nu_e S_e^2}{2\sigma_e^2} \right\}. \end{aligned}$$

The posterior distribution for $\boldsymbol{\beta}$ can be written as

$$\begin{aligned}
f(\beta | \mathbf{y}, \sigma_\beta^2, \sigma_e^2) &= \frac{f(\mathbf{y} | \beta, \sigma_\beta^2, \sigma_e^2) f(\beta | \sigma_\beta^2) f(\sigma_e^2)}{f(\mathbf{y}, \sigma_\beta^2, \sigma_e^2)} \\
&\propto f(\mathbf{y} | \beta, \sigma_\beta^2, \sigma_e^2) f(\beta | \sigma_\beta^2) f(\sigma_e^2) \\
&\propto f(\mathbf{y} | \beta, \sigma_\beta^2, \sigma_e^2) f(\beta | \sigma_\beta^2) \\
&\propto (\sigma_e^2)^{-n/2} \exp \left\{ -\frac{(\mathbf{y} - \mathbf{X}\beta)'(\mathbf{y} - \mathbf{X}\beta)}{2\sigma_e^2} \right\} \\
&\times (\sigma_\beta^2)^{-k/2} \exp \left\{ -\frac{\sum_{j=1}^k \beta_j^2}{2\sigma_\beta^2} \right\} \\
&\propto \exp \left\{ -\frac{(\mathbf{y} - \mathbf{X}\beta)'(\mathbf{y} - \mathbf{X}\beta) + \sum_{j=1}^k \beta_j^2 \frac{\sigma_e^2}{\sigma_\beta^2}}{2\sigma_e^2} \right\} \\
&\propto \exp \left\{ -\frac{\mathbf{y}'\mathbf{y} - 2\mathbf{y}'\mathbf{X}\beta + \beta'(\mathbf{X}'\mathbf{X} + \mathbf{D}\frac{\sigma_e^2}{\sigma_\beta^2})\beta}{2\sigma_e^2} \right\} \\
&\propto \exp \left\{ -\frac{\mathbf{y}'\mathbf{y} - (\hat{\beta} - \hat{\beta}')(\mathbf{X}'\mathbf{X} + \mathbf{D}\frac{\sigma_e^2}{\sigma_\beta^2})(\hat{\beta} - \hat{\beta}) - \hat{\beta}'(\mathbf{X}'\mathbf{X} + \mathbf{D}\frac{\sigma_e^2}{\sigma_\beta^2})\hat{\beta}}{2\sigma_e^2} \right\} \\
&\propto \exp \left\{ -\frac{(\hat{\beta} - \hat{\beta}')(\mathbf{X}'\mathbf{X} + \mathbf{D}\frac{\sigma_e^2}{\sigma_\beta^2})(\hat{\beta} - \hat{\beta})}{2\sigma_e^2} \right\},
\end{aligned}$$

for

$$(\mathbf{X}'\mathbf{X} + \mathbf{D}\frac{\sigma_e^2}{\sigma_\beta^2})\hat{\beta} = \mathbf{X}'\mathbf{y}, \quad (3)$$

where \mathbf{D} is a diagonal matrix with zero on the first diagonal and ones on the remaining diagonals. Thus, the full-conditional posterior for β is a normal distribution with mean given by (3) and variance

$$(\mathbf{X}'\mathbf{X} + \mathbf{D}\frac{\sigma_e^2}{\sigma_\beta^2})^{-1}\sigma_e^2.$$

Full-conditionals:

The full conditionals for β_0 and σ_e^2 are identical to those in simple linear regression.

Full-conditional for β_j

The full-conditional for β_j is obtained by dropping from the joint posterior all terms and factors that do not involve β_j :

$$\begin{aligned} f(\beta_j | \text{ELSE}) &\propto \exp \left\{ -\frac{(\mathbf{w}_j - \mathbf{x}_j \beta_j)' (\mathbf{w}_j - \mathbf{x}_j \beta_j)}{2\sigma_e^2} \right\} \\ &\quad \times \exp \left\{ -\frac{\beta_j^2}{2\sigma_\beta^2} \right\} \\ &\propto \exp \left\{ -\frac{\mathbf{w}'_j \mathbf{w}_j - 2\mathbf{w}'_j \mathbf{x}_j \beta_j + \beta_j^2 (\mathbf{x}'_j \mathbf{x}_j + \sigma_e^2 / \sigma_\beta^2)}{2\sigma_e^2} \right\} \\ &\propto \exp \left\{ -\frac{\mathbf{w}'_j \mathbf{w}_j - (\beta_j - \hat{\beta}_j)^2 (\mathbf{x}'_j \mathbf{x}_j + \sigma_e^2 / \sigma_\beta^2) - \hat{\beta}_j^2 (\mathbf{x}'_j \mathbf{x}_j + \sigma_e^2 / \sigma_\beta^2)}{2\sigma_e^2} \right\} \\ &\propto \exp \left\{ -\frac{(\beta_j - \hat{\beta}_j)^2}{\frac{2\sigma_e^2}{(\mathbf{x}'_j \mathbf{x}_j + \sigma_e^2 / \sigma_\beta^2)}} \right\}, \end{aligned}$$

where $\hat{\beta}_j = \frac{\mathbf{w}'_j \mathbf{x}_j}{\mathbf{x}'_j \mathbf{x}_j + \sigma_e^2 / \sigma_\beta^2}$, and $\mathbf{w}_j = \mathbf{y} - \sum_{l \neq j} \mathbf{x}_l \beta_l$. So, the full-conditional posterior for β_j is a normal distribution with mean $\hat{\beta}_j$ and variance $\frac{\sigma_e^2}{\mathbf{x}'_j \mathbf{x}_j + \sigma_e^2 / \sigma_\beta^2}$.

Exercise

1. Use $\beta_0 = 1$, $\sigma_\beta^2 = 0.1$ and $\sigma_e^2 = 1.0$ to generate a data set with 10 observations from model (2) with $k = 15$ covariates.
2. Setup and solve the mixed model equations given by (3).
3. Sample the elements of β using Gibbs.
4. Compute the posterior mean of β from the samples and compare with the mixed model solutions.
5. Compute the posterior covariance matrix from the sampled values. Compare results with inverse of the mixed-model coefficient matrix.

Model with unknown σ_β^2

In the previous section, we assumed that σ_β^2 in the prior of the slopes was known. Here, we will consider this variance to be unknown with a scaled inverted chi-square prior with scale parameter S_β^2 and degrees of freedom ν_β . The joint posterior for this model is

$$\begin{aligned} f(\theta|y) &\propto f(y|\theta)f(\theta) \\ &\propto (\sigma_e^2)^{-n/2} \exp\left\{-\frac{(y - X\beta)'(y - X\beta)}{2\sigma_e^2}\right\} \\ &\times (\sigma_\beta^2)^{-k/2} \exp\left\{-\frac{\sum_{j=1}^k \beta_j^2}{2\sigma_\beta^2}\right\} \\ &\times (\sigma_\beta^2)^{-(2+\nu_\beta)/2} \exp\left\{-\frac{\nu_\beta S_\beta^2}{2\sigma_\beta^2}\right\} \\ &\times (\sigma_e^2)^{-(2+\nu_e)/2} \exp\left\{-\frac{\nu_e S_e^2}{2\sigma_e^2}\right\}. \end{aligned}$$

Then, the full-conditional posterior for σ_β^2 is

$$\begin{aligned} f(\sigma_\beta^2|y, \beta, \sigma_e^2) &\propto (\sigma_\beta^2)^{-k/2} \exp\left\{-\frac{\sum_{j=1}^k \beta_j^2}{2\sigma_\beta^2}\right\} \\ &\times (\sigma_\beta^2)^{-(2+\nu_\beta)/2} \exp\left\{-\frac{\nu_\beta S_\beta^2}{2\sigma_\beta^2}\right\} \\ &\propto (\sigma_\beta^2)^{-(2+k+\nu_\beta)/2} \exp\left\{-\frac{\sum_{j=1}^k \beta_j^2 + \nu_\beta S_\beta^2}{2\sigma_\beta^2}\right\}, \end{aligned}$$

which can be recognized as a scaled inverted chi-square distribution with $\tilde{\nu}_\beta = k + \nu_\beta$ degrees of freedom and scale parameter $\tilde{S}_\beta^2 = (\sum_{j=1}^k \beta_j^2 + \nu_\beta S_\beta^2)/\tilde{\nu}_\beta$. A sample from this posterior can be obtained as

$$\frac{\sum_{j=1}^k \beta_j^2 + \nu_\beta S_\beta^2}{\chi_{\nu_\beta}^2}.$$

Exercise

Extend the sampler used in the previous section to treat σ_β^2 as an unknown. Plot the posterior distribution σ_β^2 .

Model with unknown covariate-specific variances

Here we consider a model where the prior for the slope corresponding to covariate j is normal with mean 0 and variance σ_j^2 , where σ_j^2 has scaled inverted chi-square prior with scale parameter S_β^2 and degrees of freedom ν_β . The joint posterior for this model is

$$\begin{aligned} f(\theta|y) &\propto f(y|\theta)f(\theta) \\ &\propto (\sigma_e^2)^{-n/2} \exp\left\{-\frac{(y - X\beta)'(y - X\beta)}{2\sigma_e^2}\right\} \\ &\times \prod_{j=1}^k (\sigma_j^2)^{-1/2} \exp\left\{-\frac{\beta_j^2}{2\sigma_j^2}\right\} \\ &\times \prod_{j=1}^k (\sigma_j^2)^{-(2+\nu_\beta)/2} \exp\left\{-\frac{\nu_\beta S_\beta^2}{2\sigma_j^2}\right\} \\ &\times (\sigma_e^2)^{-(2+\nu_e)/2} \exp\left\{-\frac{\nu_e S_e^2}{2\sigma_e^2}\right\}. \end{aligned}$$

It can be shown that:

1. The full-conditional posterior for β_j is normal with mean

$$\hat{\beta}_j = \frac{\mathbf{x}'_j \mathbf{w}_j}{(\mathbf{x}'_j \mathbf{x}_j + \sigma_e^2 / \sigma_j^2)},$$

and variance $\frac{\sigma_e^2}{(\mathbf{x}'_j \mathbf{x}_j + \sigma_e^2 / \sigma_j^2)}$.

2. The full-conditional posterior for σ_j^2 is a scaled inverted chi-square distribution with $\tilde{\nu}_\beta = 1 + \nu_\beta$ degrees of freedom and scale parameter $\tilde{S}_\beta^2 = (\beta_j^2 + \nu_\beta S_\beta^2) / \tilde{\nu}_\beta$. A sample from this posterior can be obtained as $\frac{\beta_j^2 + \nu_\beta S_\beta^2}{\chi_{\nu_\beta}^2}$.
3. Marginally, the prior for β_j is a scaled t distribution with ν_β degrees of freedom, mean 0 and scale parameter S_β^2 .

Exercise

- Derive the full-conditional posterior for β_j .
- Derive the full-conditional posterior for σ_j^2 .
- Use a Gibbs sampler to compute the posterior mean of β .

Model with Mixture Prior for Regression Coefficients

- before, a flat prior is used for the intercept, μ . The prior for slope j is a mixture:

$$\beta_j = \begin{cases} 0 & \text{probability } \pi \\ \sim N(0, \sigma_\beta^2) & \text{probability } (1 - \pi) \end{cases},$$

where σ_β^2 has a scaled inverted chi-square prior with scale parameter S_β^2 and degrees of freedom ν_β . In order to use the Gibbs sampler, it is convenient to write β_j as

$$\beta_j = \delta_j \gamma_j,$$

where δ_j is a Bernoulli variable with probability $1 - \pi$ of being 1:

$$\delta_j = \begin{cases} 0 & \text{probability } \pi \\ 1 & \text{probability } (1 - \pi) \end{cases},$$

and γ_j is normally distributed with mean zero and variance σ_β^2 . Then, the model for the phenotypic values can be written as

$$y_i = \mu + \sum_{j=1} X_{ij} \gamma_j \delta_j + e_i.$$

Full-conditionals:

Joint posterior for all the parameters is proportional to

$$\begin{aligned} f(\theta | \mathbf{y}) &\propto f(\mathbf{y} | \theta) f(\theta) \\ &\propto (\sigma_e^2)^{-n/2} \exp \left\{ -\frac{(\mathbf{y} - \mathbf{1}\mu - \sum \mathbf{X}_j \gamma_j \delta_j)'(\mathbf{y} - \mathbf{1}\mu - \sum \mathbf{X}_j \gamma_j \delta_j)}{2\sigma_e^2} \right\} \\ &\times \prod_{j=1}^k (\sigma_\beta^2)^{-1/2} \exp \left\{ -\frac{\gamma_j^2}{2\sigma_\beta^2} \right\} \\ &\times \prod_{j=1}^k \pi^{(1-\delta_j)} (1-\pi)^{\delta_j} \\ &\times (\sigma_\beta^2)^{-(\nu_\beta+2)/2} \exp \left\{ -\frac{\nu_\beta S_\beta^2}{2\sigma_\beta^2} \right\} \\ &\times (\sigma_e^2)^{-(2+\nu_e)/2} \exp \left\{ -\frac{\nu_e S_e^2}{2\sigma_e^2} \right\}, \end{aligned}$$

where θ denotes all the unknowns.

Full-conditional for μ

full-conditional for μ is a normal distribution with mean $\hat{\mu}$ and variance $\frac{\sigma_e^2}{n}$, where $\hat{\mu}$ is the least-squares estimate of μ in the model

$$\mathbf{y} - \sum_{j=1}^k \mathbf{X}_j \gamma_j \delta_j = \mathbf{1}\mu + \mathbf{e},$$

and $\frac{\sigma_e^2}{n}$ is the variance of this estimator (n is the number of observations).

Full-conditional for γ_j

$$\begin{aligned} f(\gamma_j | \text{ELSE}) &\propto \exp \left\{ -\frac{(\mathbf{w}_j - \mathbf{X}_j \gamma_j \delta_j)' (\mathbf{w}_j - \mathbf{X}_j \gamma_j \delta_j)}{2\sigma_e^2} \right\} \\ &\times \exp \left\{ -\frac{\gamma_j^2}{2\sigma_\beta^2} \right\} \\ &\propto \exp \left\{ -\frac{[\mathbf{w}'_j \mathbf{w}_j - 2\mathbf{w}'_j \mathbf{X}_j \gamma_j \delta_j + \gamma_j^2 (\mathbf{x}'_j \mathbf{x}_j \delta_j + \sigma_e^2 / \sigma_\beta^2)]}{2\sigma_e^2} \right\} \\ &\propto \exp \left\{ -\frac{(\gamma_j - \hat{\gamma}_j)^2}{\frac{2\sigma_e^2}{(\mathbf{x}'_j \mathbf{x}_j \delta_j + \sigma_e^2 / \sigma_\beta^2)}} \right\}, \end{aligned}$$

where

$$\mathbf{w}_j = \mathbf{y} - \mathbf{1}\mu - \sum_{l \neq j} \mathbf{X}_l \gamma_l \delta_l.$$

So, the full-conditional for γ_j is a normal distribution with mean

$$\hat{\gamma}_j = \frac{\mathbf{X}'_j \mathbf{w}_j \delta_j}{(\mathbf{x}'_j \mathbf{x}_j \delta_j + \sigma_e^2 / \sigma_\beta^2)}$$

and variance $\frac{\sigma_e^2}{(\mathbf{x}'_j \mathbf{x}_j \delta_j + \sigma_e^2 / \sigma_\beta^2)}$.

Full-conditional for δ_j

$$\Pr(\delta_j = 1 | \text{ELSE}) \propto \frac{h(\delta_j = 1)}{h(\delta_j = 1) + h(\delta_j = 0)},$$

where $h(\delta_j) = \pi^{(1-\delta_j)}(1-\pi)^{\delta_j} \exp \left\{ -\frac{(\mathbf{w}_j - \mathbf{X}_j \gamma_j \delta_j)' (\mathbf{w}_j - \mathbf{X}_j \gamma_j \delta_j)}{2\sigma_e^2} \right\}$.

Full-conditional for σ_β^2

$$\begin{aligned}
f(\sigma_\beta^2 | \text{ELSE}) &\propto (\sigma_\beta^2)^{-k/2} \exp \left\{ -\frac{\sum_{j=1}^k \gamma_j^2}{2\sigma_\beta^2} \right\} \\
&\times (\sigma_\beta^2)^{-(\nu_\beta+2)/2} \exp \left\{ -\frac{\nu_\beta S_\beta^2}{2\sigma_\beta^2} \right\} \\
&\propto (\sigma_\beta^2)^{-(k+\nu_\beta+2)/2} \exp \left\{ -\frac{\sum_{j=1}^k \gamma_j^2 + \nu_\beta S_\beta^2}{2\sigma_\beta^2} \right\},
\end{aligned}$$

and this is proportional to a scaled inverted chi-square distribution with $\tilde{\nu}_\beta = \nu_\beta + k$ and scale parameter $\tilde{S}_\beta^2 = (\sum_{j=1}^k \gamma_j^2 + \nu_\beta S_\beta^2)/\tilde{\nu}_\beta$.

Full-conditional for π

$$f(\pi | \text{ELSE}) \propto \pi^{(k - \sum_{j=1}^k \delta_j)} (1 - \pi)^{\sum_{j=1}^k \delta_j},$$

which is proportional to a Beta distribution with parameters $a = k - \sum_{j=1}^k \delta_j + 1$ and $b = \sum \delta_j + 1$.

Full-conditional for σ_e^2

$$\begin{aligned}
f(\sigma_e^2 | \text{ELSE}) &\propto (\sigma_e^2)^{-n/2} \exp \left\{ -\frac{(\mathbf{y} - \mathbf{1}\mu - \sum \mathbf{X}_j \gamma_j \delta_j)'(\mathbf{y} - \mathbf{1}\mu - \sum \mathbf{X}_j \gamma_j \delta_j)}{2\sigma_e^2} \right\} \\
&\times (\sigma_e^2)^{-(2+\nu_e)/2} \exp \left\{ -\frac{\nu_e S_e^2}{2\sigma_e^2} \right\} \\
&\propto (\sigma_e^2)^{-(n+2+\nu_e)/2} \exp \left\{ -\frac{(\mathbf{y} - \mathbf{1}\mu - \sum \mathbf{X}_j \gamma_j \delta_j)'(\mathbf{y} - \mathbf{1}\mu - \sum \mathbf{X}_j \gamma_j \delta_j) + \nu_e S_e^2}{2\sigma_e^2} \right\},
\end{aligned}$$

which is proportional to a scaled inverted chi-square density with $\tilde{\nu}_e = n + \nu_e$ degrees of freedom and $\tilde{S}_e^2 = \frac{(\mathbf{y} - \mathbf{1}\mu - \sum \mathbf{X}_j \gamma_j \delta_j)'(\mathbf{y} - \mathbf{1}\mu - \sum \mathbf{X}_j \gamma_j \delta_j) + \nu_e S_e^2}{\tilde{\nu}_e}$ scale parameter.

Bayesian Inference by Application to Simple Linear Regression

Simple linear regression is used to illustrate Bayesian inference, using the Gibbs sampler. The Gibbs sampler is used to draw samples from the posterior distribution of the intercept, the slope and the residual variance.

The Model

Consider the linear model:

$$y_i = \beta_0 + x_i\beta_1 + e_i. \quad (35)$$

where for observation i , y_i is the value of the dependent variable, β_0 is the intercept, x_i is the value of the independent variable and e_i is a residual. Flat priors are used for the intercept and slope, and the residuals are assumed to be identically and independently distributed normal random variables with mean zero and variance σ_e^2 . A scaled inverted chi-square prior is used for σ_e^2 .

Simulation of Data

In [1]:

```
using Distributions
using StatsBase
```

In [20]:

```
n = 20 #number of observations
k = 1 #number of covariates

x = sample([0,1,2],(n,k))
X = hcat(ones(Int64,n),x)

betaTrue = [1,2]
y = X*betaTrue+ randn(n);
```

Least Squares Estimation

In matrix notation, the model (35) is

$$\mathbf{y} = \mathbf{X}\beta + \mathbf{e},$$

where

$$\mathbf{X} = \begin{bmatrix} 1 & x_1 & 1 & x_2 & \vdots & \vdots & 1 & x_n \end{bmatrix}.$$

Then, the least-squares estimator of β is

$$\hat{\beta} = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{y},$$

and the variance of this estimator is

$$Var(\hat{\beta}) = (\mathbf{X}'\mathbf{X})^{-1}\sigma_e^2.$$

Calculations in Julia:

In [3]:

```
XPX = X'*X
rhs = X'*y
XPXi= inv(XPX)
println(XPXi)

16363636363636364 -0.09090909090909091
-0.090909090909091 0.07272727272727274]
```

In [4]:

```
betaHat = XPXi*rhs
println(betaHat)

[0.6986138506616033, 2.293983905821345]
```

In [5]:

```
eHat = y - X*betaHat
resVar = eHat'*eHat/(n-2)
println(resVar)

[0.45974834730130465]
```

Bayesian Inference

Consider making inferences about β from $f(\beta|y, \sigma_e^2)$. By using the Bayes theorem, this conditional density is given as

$$\begin{aligned} f(\beta|y, \sigma_e^2) &= \frac{f(y|\beta, \sigma_e^2)f(\beta)f(\sigma_e^2)}{f(y, \sigma_e^2)} \\ &\propto f(y|\beta, \sigma_e^2)f(\beta)f(\sigma_e^2) \\ &\propto f(y|\beta, \sigma_e^2) \\ &= (2\pi\sigma_e^2)^{-n/2} \exp \left\{ -\frac{1}{2} \frac{(y - X\beta)'(y - X\beta)}{\sigma_e^2} \right\} \end{aligned} \quad (36)$$

which looks like the n -dimensional normal density of y with mean $X\beta$ and covariance matrix $I\sigma_e^2$. But, $f(\beta|y, \sigma_e^2)$ should be a two-dimensional density. So, the quadratic $Q = (y - X\beta)'(y - X\beta)$ in the exponent of (36) is rearranged as

$$\begin{aligned} Q &= (y - X\beta)'(y - X\beta) \\ &= y'y - 2y'X\beta + \beta'(X'X)\beta \\ &= y'y + (\beta - \hat{\beta})'(X'X)(\beta - \hat{\beta}) - \hat{\beta}'(X'X)\hat{\beta}, \end{aligned}$$

where $\hat{\beta}$ is the solution to $(X'X)\hat{\beta} = X'y$, which is the least-squares estimator of β . In this expression, only the second term depends on β . Thus, $f(\beta|y, \sigma_e^2)$ can be written as

$$f(\beta|y, \sigma_e^2) \propto \exp \left\{ -\frac{1}{2} \frac{(\beta - \hat{\beta})'(X'X)(\beta - \hat{\beta})}{\sigma_e^2} \right\},$$

which can be recognized as proportional to the density for a two-dimensional normal distribution with mean $\hat{\beta}$ and variance $(X'X)^{-1}\sigma_e^2$. Thus, in this simple setting, the posterior mean of β is given by the least-squares estimate, and drawing samples from the posterior are not needed. But, to illustrate the Gibbs sampler, we will apply it to this simple example.

Gibbs Sampler for β

The simple regression model can be written as

$$\mathbf{y} = \mathbf{1}\beta_0 + \mathbf{x}\beta_1 + \mathbf{e}.$$

In the Gibbs sampler, β_0 is sampled from its full-conditional posterior: $f(\beta_0 | \mathbf{y}, \beta_1, \sigma_e^2)$. This conditional distribution is computed for the current values of β_1 and σ_e^2 . So, we can write the model as

$$\mathbf{w}_0 = \mathbf{1}\beta_0 + \mathbf{e},$$

where $\mathbf{w}_0 = \mathbf{y} - \mathbf{x}\beta_1$. Then, the least-squares estimator of β_0 is

$$\hat{\beta}_0 = \frac{\mathbf{1}' \mathbf{w}_0}{\mathbf{1}' \mathbf{1}},$$

and the variance of this estimator is

$$Var(\hat{\beta}_0) = \frac{\sigma_e^2}{\mathbf{1}' \mathbf{1}}.$$

By applying the strategy used to derive $f(\beta_1 | \mathbf{y}, \sigma_e^2)$ above, the full-conditional posterior for β_0 can be shown to be a normal distribution with mean $\hat{\beta}_0$ and variance $\frac{\sigma_e^2}{\mathbf{1}' \mathbf{1}}$. Similarly, the full-conditional posterior for β_1 is a normal distribution with mean

$$\hat{\beta}_1 = \frac{\mathbf{x}' \mathbf{w}_1}{\mathbf{x}' \mathbf{x}}$$

and variance $\frac{\sigma_e^2}{\mathbf{x}' \mathbf{x}}$, where $\mathbf{w}_1 = \mathbf{y} - \mathbf{1}\beta_0$. In the calculations below, we will use the true value of σ_e^2 .

Calculations in Julia:

In [9]:

```
# loop for Gibbs sampler
niter = 10000 # number of samples
    = [0.0, 0.0]
meanB = [0.0, 0.0]
a=Float64[ ]

for iter = 1:niter

    # sampling intercept
    w = y - X[:,2] * b[2]
    x = X[:,1]
    xpxi = 1/(x'*x)[1,1]
    bHat = (xpxi*x'*w)[1,1]
    b[1] = rand(Normal(bHat, sqrt(xpxi))) # using residual var = 1

    # sampling slope
    w = y - X[:,1]*b[1]
    x = X[:,2]
    xpxi = 1/(x'*x)[1,1]
    bHat = (xpxi*x'*w)[1,1]
    b[2] = rand(Normal(bHat, sqrt(xpxi))) # using residual var = 1
    meanB = meanB + b
    push!(a,b[2])

if ((iter%1000) == 0)
    @printf("Intercept = %6.3f \n", meanB[1]/iter)
    @printf("Slope      = %6.3f \n", meanB[2]/iter)
end
end
```

```
Intercept = 0.725
Slope     = 2.283
Intercept = 0.695
Slope     = 2.301
    ercept = 0.700
    slope   = 2.297
Intercept = 0.702
Slope     = 2.294
Intercept = 0.700
Slope     = 2.294
Intercept = 0.696
Slope     = 2.296
Intercept = 0.699
Slope     = 2.294
Intercept = 0.709
Slope     = 2.287
Intercept = 0.714
Slope     = 2.283
Intercept = 0.712
Slope     = 2.285
```

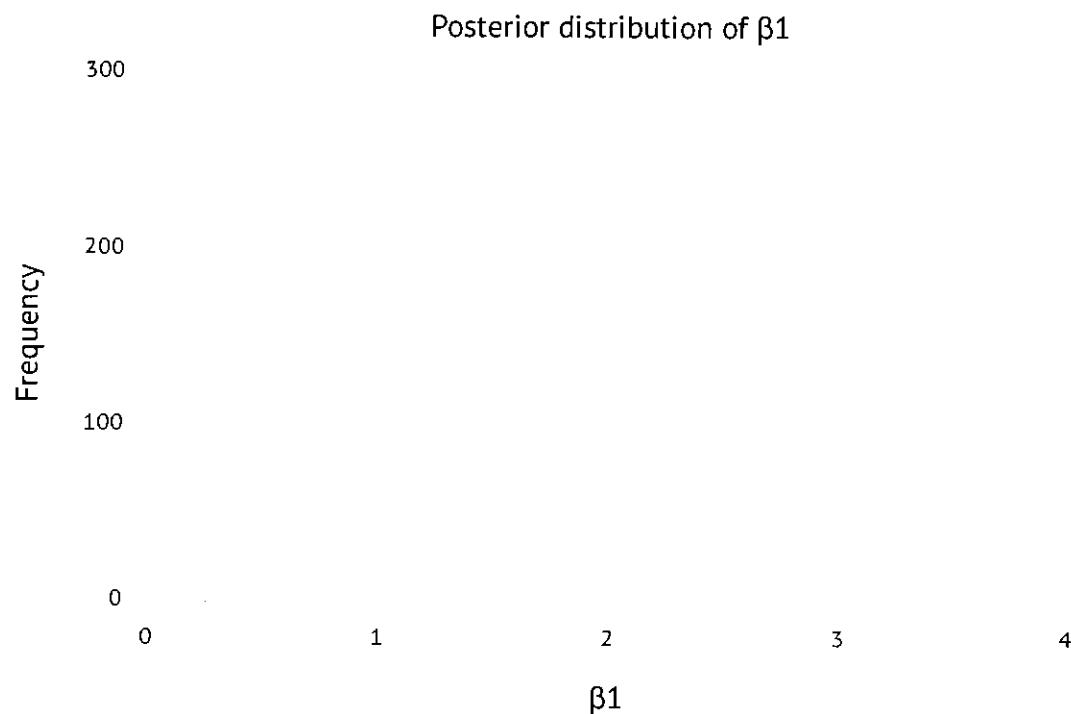
In [11]:

```
using Gadfly
```

In [15]:

```
plot(x=a, Geom.histogram,  
Guide.title("Posterior distribution of β1"),  
      de.ylabel("Frequency"),  
      de.xlabel("β1"))
```

Out[15]:



Full-conditional Posterior for σ_e^2

recall that we assumed a scaled inverted chi-square prior for σ_e^2 . The density function for this is:

$$f(\sigma_e^2) = \frac{(S_e^2 \nu_e / 2)^{\nu_e/2}}{\Gamma(\nu_e/2)} (\sigma_e^2)^{-(2+\nu_e)/2} \exp\left\{-\frac{\nu_e S_e^2}{2\sigma_e^2}\right\}, \quad (37)$$

where S_e^2 and ν_e are the scale and the degrees of freedom parameters for this distribution. Applying Bayes theorem to combine this prior with the “likelihood” (given in (36)), the full-conditional posterior for the residual variance can be written as

$$\begin{aligned} f(\sigma_e^2 | \mathbf{y}, \boldsymbol{\beta}) &= \frac{f(\mathbf{y}|\boldsymbol{\beta}, \sigma_e^2) f(\boldsymbol{\beta}) f(\sigma_e^2)}{f(\mathbf{y}, \boldsymbol{\beta})} \\ &\propto f(\mathbf{y}|\boldsymbol{\beta}, \sigma_e^2) f(\boldsymbol{\beta}) f(\sigma_e^2) \\ &\propto (\sigma_e^2)^{-n/2} \exp\left\{-\frac{1}{2} \frac{(\mathbf{y} - \mathbf{X}\boldsymbol{\beta})'(\mathbf{y} - \mathbf{X}\boldsymbol{\beta})}{\sigma_e^2}\right\} \\ &\times (\sigma_e^2)^{-(2+\nu_e)/2} \exp\left\{-\frac{\nu_e S_e^2}{2\sigma_e^2}\right\} \\ &= (\sigma_e^2)^{-(n+2+\nu_e)/2} \exp\left\{-\frac{(\mathbf{y} - \mathbf{X}\boldsymbol{\beta})'(\mathbf{y} - \mathbf{X}\boldsymbol{\beta}) + \nu_e S_e^2}{2\sigma_e^2}\right\}. \end{aligned} \quad (38)$$

Comparing (38) with (37), can see that it is proportional to a scaled inverse chi-squared density with

$= n + \nu_e$ degrees of freedom and $\tilde{S}_e^2 = \frac{(\mathbf{y} - \mathbf{X}\boldsymbol{\beta})'(\mathbf{y} - \mathbf{X}\boldsymbol{\beta}) + \nu_e S_e^2}{\tilde{\nu}_e}$ scale parameter. A sample from this density can be obtained as $\frac{(\mathbf{y} - \mathbf{X}\boldsymbol{\beta})'(\mathbf{y} - \mathbf{X}\boldsymbol{\beta}) + \nu_e S_e^2}{\chi_{\tilde{\nu}_e}^2}$, where $\chi_{\tilde{\nu}_e}^2$ is a chi-squared random variable with $\tilde{\nu}_e$ degrees of freedom.

Exercise

In the Julia script given here, the simulated value of the residual variance was used in the sampling of $\boldsymbol{\beta}$. Extend this script to also sample σ_e^2 from its full-conditional posterior given above. In Julia, `rand(Chisq(ν),1)` gives a chi-squared random variable with ν degrees of freedom. Solutions can be found [here](#) ([./solutions/BayesSimpleLinearExercise.ipynb](#)) where flat priors for σ_e^2 is used.

Model with Normal Prior for Slope

Consider the simple regression model that can be written as

$$\mathbf{y} = \mathbf{1}\beta_0 + \mathbf{x}\beta_1 + \mathbf{e}.$$

Here we consider a model with a flat prior for β_0 and a normal prior for the slope:

$$\beta_1 \sim N(0, \sigma_\beta^2),$$

where σ_β^2 is assumed to be known.

Then, the full-conditional posterior for $\theta' = [\beta, \sigma_e^2]$ is

$$\begin{aligned} f(\theta|y) &\propto f(y|\theta)f(\theta) \\ &\propto (\sigma_e^2)^{-n/2} \exp \left\{ -\frac{(y - \mathbf{1}\beta_0 - \mathbf{x}\beta_1)'(y - \mathbf{1}\beta_0 - \mathbf{x}\beta_1)}{2\sigma_e^2} \right\} \\ &\quad \times (\sigma_\beta^2)^{-1/2} \exp \left\{ -\frac{\beta_1^2}{2\sigma_\beta^2} \right\} \\ &\quad \times (\sigma_e^2)^{-(2+\nu_e)/2} \exp \left\{ -\frac{\nu_e S_e^2}{2\sigma_e^2} \right\}. \end{aligned}$$

Full-conditional for β_1 :

The full-conditional for β_1 is obtained by dropping all terms and factors that do not involve β_1 :

$$\begin{aligned} f(\beta_1|ELSE) &\propto \exp \left\{ -\frac{(y - \mathbf{1}\beta_0 - \mathbf{x}\beta_1)'(y - \mathbf{1}\beta_0 - \mathbf{x}\beta_1)}{2\sigma_e^2} \right\} \times \exp \left\{ -\frac{\beta_1^2}{2\sigma_\beta^2} \right\} \\ &\propto \exp \left\{ -\frac{\mathbf{w}'\mathbf{w} - 2\mathbf{w}'\mathbf{x}\beta_1 + \beta_1^2(\mathbf{x}'\mathbf{x} + \sigma_e^2/\sigma_\beta^2)}{2\sigma_e^2} \right\} \\ &\propto \exp \left\{ -\frac{\mathbf{w}'\mathbf{w} - (\beta_1 - \hat{\beta}_1)^2(\mathbf{x}'\mathbf{x} + \sigma_e^2/\sigma_\beta^2) - \hat{\beta}_1^2(\mathbf{x}'\mathbf{x} + \sigma_e^2/\sigma_\beta^2)}{2\sigma_e^2} \right\} \\ &\propto \exp \left\{ -\frac{(\beta_1 - \hat{\beta}_1)^2}{\frac{2\sigma_e^2}{(\mathbf{x}'\mathbf{x} + \sigma_e^2/\sigma_\beta^2)}} \right\}, \end{aligned}$$

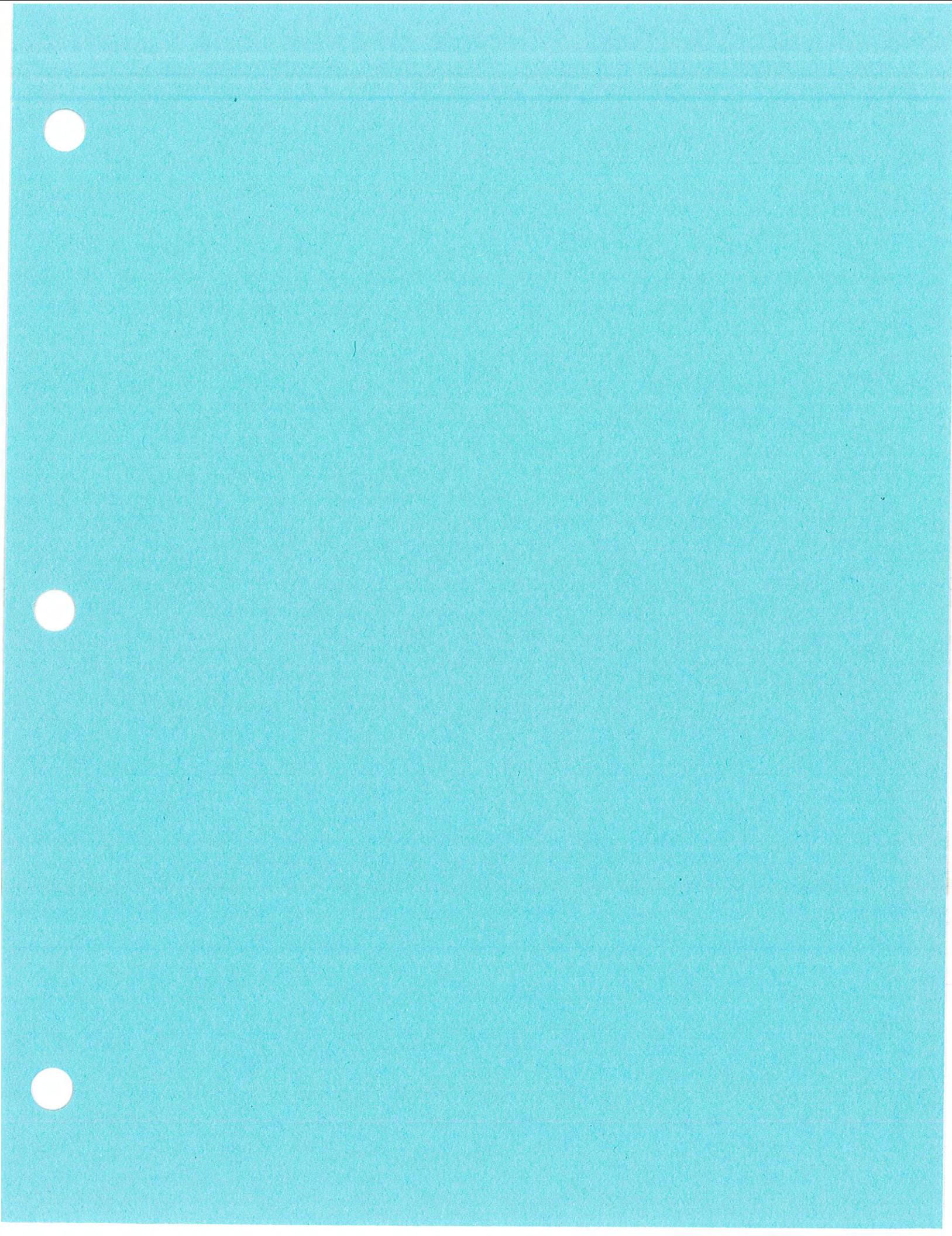
where

$$\hat{\beta}_1 = \frac{\mathbf{x}'\mathbf{w}}{(\mathbf{x}'\mathbf{x} + \sigma_e^2/\sigma_\beta^2)},$$

and $\mathbf{w} = \mathbf{y} - \mathbf{1}\beta_0$. So, the full-conditional posterior for β_1 is a normal distribution with mean $\hat{\beta}_1$ and variance $\frac{\sigma_e^2}{(\mathbf{x}'\mathbf{x} + \sigma_e^2/\sigma_\beta^2)}$.

Exercise

1. Use Julia to simulate a vector of 1000 values for β_1 from a normal distribution with mean zero and variance 3. Plot a histogram of these values.
2. Use $\beta_0 = 1$, $\beta_1 = 2$ and $\sigma_e^2 = 5$, to generate a vector of observations, y , that follows a simple linear regression model.
3. Use the Gibbs sampler to draw 10,000 samples for β_1 from its posterior distribution.
 - A. Compute the mean and variance of the sampled values.
 - B. Draw a histogram of the sampled values. Compare with prior.



An Equivalent (animal) Model for Genomic Prediction

More loci than animals

Allelic effects – but for selection we are more interested in animal (not allelic) merit

$$\begin{aligned} \mathbf{y} &= \mathbf{l}\boldsymbol{\mu} + \sum_{i=1}^{t \text{ plus } 1} \mathbf{M}_i \mathbf{a}_i + \mathbf{e} \\ \mathbf{y} &= \mathbf{l}\boldsymbol{\mu} + \mathbf{l} \left\{ \sum_{i=1}^{t \text{ plus } 1} \mathbf{M}_i \mathbf{a}_i \right\} + \mathbf{e} \\ \mathbf{y} &= \mathbf{l}\boldsymbol{\mu} + "Z""\mathbf{u}" + \mathbf{e} \end{aligned}$$

Order of MME is number of fixed effects plus number of animals
Consider the implications for 100-1,000 animals with 50,000 loci

Mixed Model Equations

$$\mathbf{y} = \mathbf{l}\boldsymbol{\mu} + \mathbf{Z}\mathbf{a} + \mathbf{e}$$

$$\begin{bmatrix} N & \mathbf{l}'\mathbf{Z} \\ \mathbf{Z}'\mathbf{l} & \mathbf{Z}'\mathbf{Z} + \sigma_e^2 \mathbf{G}^{-1} \end{bmatrix} \begin{bmatrix} \hat{\boldsymbol{\mu}} \\ \hat{\mathbf{a}} \end{bmatrix} = \begin{bmatrix} \mathbf{l}'\mathbf{y} \\ \mathbf{Z}'\mathbf{y} \end{bmatrix}, \text{ for full rank } \mathbf{G} = \text{var}(\mathbf{u})$$

$$\mathbf{y} = \mathbf{l}\boldsymbol{\mu} + \mathbf{l} \sum \mathbf{M}_i \mathbf{a}_i + \mathbf{e}$$

$$\begin{bmatrix} N & \mathbf{l}' \\ 1 & 1 + \sigma_e^2 [\text{var}(\sum \mathbf{M}_i \mathbf{a}_i)]^{-1} \end{bmatrix} \begin{bmatrix} \hat{\boldsymbol{\mu}} \\ \widehat{\sum \mathbf{M}_i \mathbf{a}_i} \end{bmatrix} = \begin{bmatrix} \mathbf{l}'\mathbf{y} \\ \mathbf{y} \end{bmatrix}$$

Order of MME is number of fixed effects plus number of animals
Consider the implications for 100-1,000 animals with 50,000 loci

Mixed Model Equations

$$\mathbf{y} = \mathbf{l}\boldsymbol{\mu} + \mathbf{l} \sum \mathbf{M}_i \mathbf{a}_i + \mathbf{e}$$

$$\begin{bmatrix} N & \mathbf{l}' \\ 1 & 1 + \sigma_e^2 [\text{var}(\sum \mathbf{M}_i \mathbf{a}_i)]^{-1} \end{bmatrix} \begin{bmatrix} \hat{\boldsymbol{\mu}} \\ \widehat{\sum \mathbf{M}_i \mathbf{a}_i} \end{bmatrix} = \begin{bmatrix} \mathbf{l}'\mathbf{y} \\ \mathbf{y} \end{bmatrix}$$

$$\text{var}(\sum \mathbf{M}_i \mathbf{a}_i) = \sum \text{var}(\mathbf{M}_i \mathbf{a}_i) = \sum \mathbf{M}_i \mathbf{A}_i \mathbf{M}_i' = \sum \mathbf{M}_i \mathbf{M}_i' \sigma_a^2 = h^2 \mathbf{A} \sigma_a^2$$

numerator relationship matrix = \mathbf{A}

$$\begin{bmatrix} N & \mathbf{l}' \\ 1 & 1 + \sigma_e^2 [\sum \mathbf{M}_i \mathbf{M}_i' \sigma_a^2]^{-1} \end{bmatrix} \begin{bmatrix} \hat{\boldsymbol{\mu}} \\ \widehat{\sum \mathbf{M}_i \mathbf{a}_i} \end{bmatrix} = \begin{bmatrix} \mathbf{l}'\mathbf{y} \\ \mathbf{y} \end{bmatrix}$$

GBLUP

- If the variance parameters are assumed known and the inverse of the genomic relationship matrix is multiplied by (known) λ , the system is known as GBLUP, as opposed to conventional pedigree or PBLUP
 - It is effectively weighting all the loci equally
 - It is similar to BayesC0 except that in that method we estimate the variance components after including a prior distribution for them

Lack of Equivalence

- The GBLUP and Marker Effects Models (MEM) such as BayesC0 with high df for the prior variances will give the same EBV for the genotyped animals
 - This is true regardless of
 - whether the models fit the A allele at every locus, the B allele at every locus, or both alleles at every locus
 - how the alleles are centered (coded 0,1,2 or -1,0,1 etc)
 - However, the PEV (and reliability) for GBLUP are not invariant to these alternative models

Genomic Analysis Combining Genotyped and Non-Genotyped Individuals

Why a Combined Analysis?

- To exploit all the available phenotypic data in GWAS and genomic prediction
 - Not just the records on genotyped individuals
 - Account for preselection of genotyped individuals
- To ensure that genomic predictions include all available information
- To avoid approximations required in multi-step analyses (that lead to double-counting)

Multi-step Genomic Prediction Analysis

- Mixed model evaluation using all phenotypes and pedigree information to generate EBV and R²
- Deregression of EBV on genotyped individuals using EBV and R² of trios of every genotyped individual, its sire and its dam
- Weighted multiple regression analysis of deregressed EBV to estimate SNP effects
- Genomic prediction DGV of genotyped individuals
- Pedigree prediction of DGV for nongenotyped
- Selection Index blending of DGV & EBV for GE-EBV

Pedigree Prediction

$$\begin{bmatrix} y_n \\ y_g \end{bmatrix} = \begin{bmatrix} X_n \\ X_g \end{bmatrix} b + \begin{bmatrix} Z_n & 0 \\ 0 & Z_g \end{bmatrix} \begin{bmatrix} u_n \\ u_g \end{bmatrix} + \begin{bmatrix} e_n \\ e_g \end{bmatrix}$$

with

$$var \begin{bmatrix} u_n \\ u_g \end{bmatrix} = \begin{bmatrix} A_{nn} & A_{ng} \\ A_{gn} & A_{gg} \end{bmatrix} \sigma_a^2.$$

Where A is the numerator relationship matrix (from pedigree) with subscripts n=non-genotyped & g=genotyped

Nejati-Javaremi et al (1997)

Replace A with $G = \sum_{i=1}^{i=\#loci} \sum_{j=1}^{j=\#alleles} m_{ij} m_{ij}'$ for genotyped

Various other authors expanded this with various approaches to center the marker covariates to create a Genomic Relationship Matrix

Fitting G^{-1} in the mixed model equations is known as GBLUP and gives the same estimates of genomic merit as MHG "BLUP"

Genotyped Animals

$$y_g = X_g b + Z_g u_g + e_g$$

Meuwissen, Hayes & Goddard (2001)

$$\text{with } u_g = M_g \alpha = \sum_{j=1}^{j=\#loci} m_j \alpha_j \delta_j$$

α_j = substitution effect

δ_j = (0,1) indicator variable

Bayesian Alphabet

$\delta_i = 1, \sigma_{\alpha}^2 = (\text{known}) \sigma_{\alpha}^2$ was "BLUP"

$\delta_i = 1, \sigma_{\alpha}^2 = (\text{unknown}) \sigma_{\alpha}^2$ was BayesA

$\delta_i = 0 \text{ with known probability } \pi$

$\sigma_{\alpha}^2 = (\text{unknown}) \sigma_{\alpha}^2$ was BayesB

$\delta_i = 0 \text{ with (un)known probability } \pi$

$\sigma_{\alpha}^2 = (\text{unknown}) \sigma_{\alpha}^2$ was BayesC or (BayesC π)

Meuwissen, Hayes & Goddard (2001)

Kizilkaya et al (2010); Habier et al (2011)

Evolution of "The Model"

Genomic Relationship Matrix

$y = u + \alpha + e$

$M = E(uu') + (I - 2E(u))\alpha\alpha' + Mee'$

$\alpha = MM' + (I - 2M)(I - M)/A$

$\alpha = (I - A)^{-1}e$

Non-Additive effects

Breeding Value Model

$y = u + \alpha + e$

$var[u] = I\sigma_u^2, var[\alpha] = I\sigma_{\alpha}^2$

$var[e] = I\sigma_e^2$

Strandor & Garrick (2009)

Equivalent

$var[u] = var[M\alpha] = MM'\sigma_{\alpha}^2$

$\alpha = M\alpha = \text{out of substitution effect}$

$y = M\alpha + e$

$var[u] = I\sigma_u^2, var[\alpha] = I\sigma_{\alpha}^2$

Meuwissen et al (2001)

What to do with the non-genotyped?

Known as Single-Step "First Attempt"

$$\text{var} \begin{bmatrix} u_n \\ u_g \end{bmatrix} = \begin{bmatrix} A_{nn} & A_{ng} \\ A_{gn} & G_{gg} \end{bmatrix} \sigma^2$$

Just replace that part of the numerator relationship matrix with genomic relationships

Then need a "brute-force" inversion of the var-cov matrix

Miszta et al (2009)

What to do with the non-genotyped?

Known as Single-Step "Second Attempt" (with brute force inverse)

$$H = \text{var} \begin{bmatrix} u_n \\ u_g \end{bmatrix} \sigma^2 = \begin{bmatrix} A_{nn} + A_{nn}A_{qq}^{-1}G_{qq}A_{qq}^{-1}A_{nn} & A_{nn}A_{qq}^{-1}G_{qq} \\ G_{qq}A_{qq}^{-1}A_{nn} & G_{qq} \end{bmatrix}$$

Legarra et al (2009)

Then with recognition of its simply structured inverse

$$H^{-1} = A^{-1} + \begin{bmatrix} 0 & 0 \\ 0 & G_{qq}^{-1} - A_{qq}^{-1} \end{bmatrix}$$

Aguirre et al (2010)

Offering programming appeal by simply replacing A^{-1} in MME by H^{-1} : known as Single-Step GBLUP and variants of which are widely used

What's wrong with Single-Step GBLUP?

- When there are less loci than genotyped individuals, G is singular
- When there are more loci than genotyped individuals, G is singular if locus covariates are centered by allele frequency
(since $G=MM'$ and $M'1=0$ then $G1=0$)
- These problems can be overcome by adhoc regression of G towards A

What's wrong with Single-Step GBLUP?

- The var-cov matrix involves a blending of A and G requiring that they represent the same "base"
 - The base in A is the pedigree founders but the allele frequencies are not usually known in that population
- It is not clear what to use to center locus covariates in populations of mixed breeds, or populations with variable breed percentages

What's wrong with Single-Step GBLUP?

- Its predictive ability can be improved by introducing another ad hoc constant κ whose optimal value can be found by trial and error

$$H^{-1} = A^{-1} + \begin{bmatrix} 0 & 0 \\ 0 & \kappa(G_{gg}^{-1} - A_{gg}^{-1}) \end{bmatrix}$$

What's wrong with Single-Step GBLUP?

- It requires brute force inversion of 2 matrices whose order is the number of genotyped individuals (ie G and A_{gg})
 - The inversion effort increase rapidly with number of genotyped individuals
 - Inversion is impractical beyond say 100,000 individuals

What's wrong with Single-Step GBLUP?

- It is not computationally straightforward for extension to Single-Step BayesA
- It is not suitable for application of mixture models (BayesB, BayesC, BayesC π)
 - But these models that provide variable selection are particularly appealing in fine-mapping applications such as with imputed NGS genotypes

Let's revisit the basic idea

$$\begin{bmatrix} y_n \\ y_g \end{bmatrix} = \begin{bmatrix} X_n \\ X_g \end{bmatrix} b + \begin{bmatrix} Z_n & 0 \\ 0 & Z_g \end{bmatrix} \begin{bmatrix} u_n \\ u_g \end{bmatrix} + \begin{bmatrix} e_n \\ e_g \end{bmatrix}$$

with $u_n = M_n \alpha$ for genotyped individuals
 whereas $u_n = \widehat{u}_n / u_n + (u_n - \widehat{u}_n / u_n) = \widehat{u}_n / u_n + \varepsilon_n$
 with $\widehat{u}_n / u_n = A_{ng} A_{gg}^{-1} u_g$
 so $u_n = A_{ng} A_{gg}^{-1} u_g + (u_n - A_{ng} A_{gg}^{-1} u_g)$

Substituting these results gives

$$\begin{aligned} \begin{bmatrix} y_n \\ y_g \end{bmatrix} &= \begin{bmatrix} X_n \\ X_g \end{bmatrix} b + \begin{bmatrix} Z_n & 0 \\ 0 & Z_g \end{bmatrix} \begin{bmatrix} u_n \\ u_g \end{bmatrix} + \begin{bmatrix} e_n \\ e_g \end{bmatrix} \\ &= \begin{bmatrix} X_n \\ X_g \end{bmatrix} b + \begin{bmatrix} Z_n & 0 \\ 0 & Z_g \end{bmatrix} \begin{bmatrix} A_{ng} A_{gg}^{-1} M_g \alpha \\ M_g \alpha \end{bmatrix} + \begin{bmatrix} Z_n & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \varepsilon_n \\ 0 \end{bmatrix} + \begin{bmatrix} e_n \\ e_g \end{bmatrix} \\ &= \begin{bmatrix} X_n \\ X_g \end{bmatrix} b + \begin{bmatrix} Z_n A_{ng} A_{gg}^{-1} M_g \\ Z_g M_g \end{bmatrix} \alpha + \begin{bmatrix} Z_n \\ 0 \end{bmatrix} \varepsilon_n + \begin{bmatrix} e_n \\ e_g \end{bmatrix} \end{aligned}$$

Fernando et al (2014) GSE

With "Hybrid" Mixed Model Equations

$$\begin{bmatrix} X'X & X'ZM & X'_g Z_n \\ M'ZX & M'Z'ZM + \psi & M'_g Z'_n Z_n \\ Z_n X_n & Z'_n Z_n M_n & Z_n Z_n + A^{-1} \lambda \end{bmatrix} \begin{bmatrix} b \\ \alpha \\ \varepsilon_n \end{bmatrix} = \begin{bmatrix} X'y \\ M'Z'y \\ Z'_n y_n \end{bmatrix}$$

where $X = \begin{bmatrix} X_n \\ X_g \end{bmatrix}$, $Z = \begin{bmatrix} Z_n & 0 \\ 0 & Z_g \end{bmatrix}$, $M = \begin{bmatrix} M_n \\ M_g \end{bmatrix} = \begin{bmatrix} A_{ng} A_{gg}^{-1} M_g \\ M_g \end{bmatrix}$, $y = \begin{bmatrix} y_n \\ y_g \end{bmatrix}$

with EBV given by
 $\widehat{u}_g = M_g \widehat{\alpha}$
 $\widehat{u}_n = M_n \widehat{\alpha} + \widehat{\varepsilon}_n$

NB Single Step GBLUP
is a special case of the above
but in this equivalent model no inversion is needed!

$$M_n = A_{ng} A_{gg}^{-1} M_g$$

If everyone is genotyped

$$\begin{bmatrix} X'X & X'ZM & X'_s Z_s \\ M'ZX & M'Z'ZM + \phi & M'_s Z'_s Z_s \\ Z_s^{-1} X_s & Z_s^{-1} Z_s M_s & Z_s^{-1} Z_s' Z_s + A'' \lambda \end{bmatrix} \begin{bmatrix} b \\ \alpha \\ \epsilon_s \end{bmatrix} = \begin{bmatrix} X'y \\ M'Zy \\ Z_s^{-1} y_s \end{bmatrix}$$

These are the MME that form the basis of BayesA, BayesB, BayesC etc

If no one is genotyped

$$\begin{bmatrix} X'X & X'ZM & X'_s Z_s \\ M'ZX & M'Z'ZM + \phi & M'_s Z'_s Z_s \\ Z_s^{-1} X_s & Z_s^{-1} Z_s M_s & Z_s^{-1} Z_s' Z_s + A''' \lambda \end{bmatrix} \begin{bmatrix} b \\ \alpha \\ \epsilon_s \end{bmatrix} = \begin{bmatrix} X'y \\ M'Zy \\ Z_s^{-1} y_s \end{bmatrix}$$

These MME form the basis of traditional pedigree-based BLUP

Invariant to Covariate Centering

Genotyped

$$\begin{aligned} y_s &= 1\mu + X_s b + Z_s M_s \alpha + e_s \\ &= 1\mu + X_s b + Z_s 1c' \alpha + Z_s (M_s - 1c') \alpha + e_s \\ \text{define } t &= c' \alpha \\ y_s &= 1(\mu + t) + X_s b + Z_s (M_s - 1c') \alpha + e_s \\ &= 1\mu + X_s b + Z_s M'_s \alpha + e_s \end{aligned}$$

.....when all animals genotyped (BayesA, BayesB etc)

But non-genotyped NOT invariant

Non-genotyped

$$\begin{aligned} y_s &= 1\mu + X_s b + Z_s A_{ss} A_s^{-1} 1c' \alpha + Z_s A_{ss} A_s^{-1} (M_s - 1c') \alpha + Z_s \epsilon_s + e_s \\ &= 1\mu + X_s b + Z_s A_{ss} A_s^{-1} 1c' \alpha + Z_s A_{ss} A_s^{-1} (M_s - 1c') \alpha + Z_s \epsilon_s + e_s \\ &= 1\mu + X_s b + Z_s A_{ss} A_s^{-1} 1t + Z_s A_{ss} A_s^{-1} M'_s \alpha + Z_s \epsilon_s + e_s \end{aligned}$$

So combined analysis of genotyped and non-genotype animals need to include a covariate for t if there is arbitrary centering (unless $t = 0$)

Computational Aspects

- It is easy to compute $A_{ss} A_{ss}^{-1} M'_s$
– And this can be done in parallel
- The computing becomes easier (rather than more difficult or impossible) as more individuals are genotyped
- Readily caters for variable selection or mixture models (eg BayesB, BayesC)
- We believe this formulation is readily extended to multi-breed and multi-trait settings
- In an MCMC framework can provide PEV

Summary

- Genomic prediction is an immature technology
- Much effort is required to extend algorithms and to develop parallel computing procedures to implement the full range of multi-breed, multi-trait, maternal effects and other models that have been routinely applied to large-scale animal prediction in recent decades

Prediction of BVs

with EBV given by

$$\widehat{u}_y = M_y \widehat{\alpha}$$

$$\widehat{u}_n = M_n \widehat{\alpha} + \widehat{\varepsilon}_n$$

or, with $M_n = A_{ng} A_{yy}^{-1} M_y$

$$\widehat{u}_n = A_{ng} A_{yy}^{-1} M_y \widehat{\alpha} + \widehat{\varepsilon}_n$$

$$= A_{ng} A_{yy}^{-1} \widehat{u}_y + \widehat{\varepsilon}_n$$