

# Bayesian whole-genome regression (Bayesian alphabet) for GWAS

Malachy Campbell

6/25/2019

# Overview

- ▶ Intro to whole genome regression
  - ▶ Dealing with  $n \ll p$
- ▶ Bayesian flavors
  - ▶ Bayesian Ridge Regression
  - ▶ Bayes A
  - ▶ Bayes B
  - ▶ Bayes C and Bayes  $C\pi$

# Quantitative genetics review

$$\mathbf{y} = \mathbf{g} + \mathbf{e}$$

## Quantitative genetics review

$$\mathbf{y} = \mathbf{g} + \mathbf{e}$$

- ▶ What is **g**?

## Quantitative genetics review

$$\mathbf{y} = \mathbf{g} + \mathbf{e}$$

$$\mathbf{y} = \sum_{j=1}^n a_j w_j + e$$

- ▶  $\mathbf{g}$  is the summation of QTL effects across all QTL for each individual

## Quantitative genetics review

$$\mathbf{y} = \mathbf{g} + \mathbf{e}$$

$$\mathbf{y} = \sum_{j=1}^n a_j w_j + e$$

- ▶  $\mathbf{g}$  is the summation of QTL effects across all QTL for each individual

## Quantitative genetics review

- ▶ With GWAS we are interested in **estimating**  $a_j$
- ▶ With genomic prediction we are interested in **predicting**  $\mathbf{g}$

## Quantitative genetics review

- ▶ With GWAS we are interested in **estimating**  $a_j$
- ▶ With genomic prediction we are interested in **predicting**  $g$
- ▶ **Why not use a model that does both?**

## Whole genome regression

$$\mathbf{y} = \mathbf{g} + \mathbf{e}$$

$$\mathbf{y} = \sum_{j=1}^n a_j w_j + e$$

- ▶ Goal is to fit all markers and obtain all  $a$ 's simultaneously

## Ordinary Least Squares (OLS)

$$\mathbf{y} = \sum_{j=1}^n a_j w_j + e$$

- ▶ Objective function for OLS

$$\hat{\mathbf{a}} = \operatorname{argmin} \left\{ \sum_i (y_i - \mu - \sum_{j=1}^p a_j w_{ij})^2 \right\}$$

## Ordinary Least Squares (OLS)

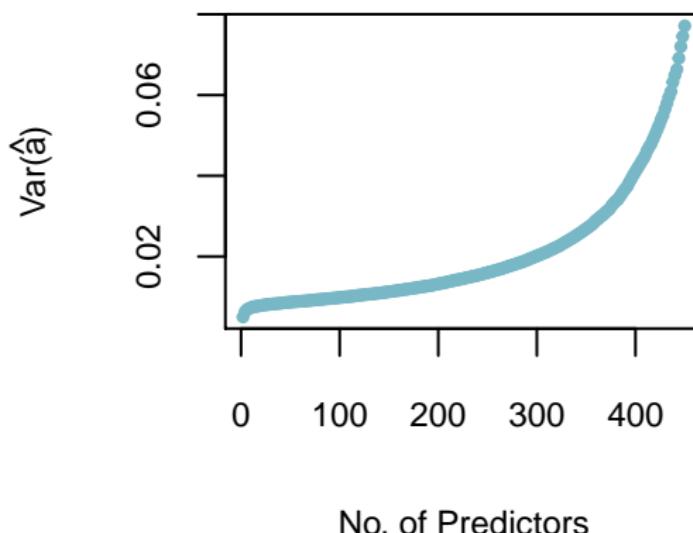
$$\mathbf{y} = \sum_{j=1}^n a_j w_j + e$$

- ▶ Objective function for OLS

$$\hat{\mathbf{a}} = \operatorname{argmin} \left\{ \underbrace{\sum_i (y_i - \mu - \sum_{j=1}^p w_{ij} a_j)^2}_{\text{residual sum of squares (RSS)}} \right\}$$

## OLS for whole genome regression

- ▶ Population size ( $n$ ) = 500
- ▶ Number of markers ( $p$ ) = 2 - 450
- ▶  $\text{Var}(\hat{\alpha}) = [\mathbf{W}'\mathbf{W}]^{-1}\sigma_e^2$ 
  - ▶  $\mathbf{W}$ :  $n \times p$
  - ▶  $\sigma_e^2 = 1$



# OLS for whole genome regression

$$\hat{\mathbf{a}} = [\mathbf{W}'\mathbf{W}]^{-1}\mathbf{W}'\mathbf{y}$$

- ▶ If  $p \gg n$

```
p <- 2000 #no of markers
w <- rbinom(prob = 0.5, n = p, size = 1) # simulate genotypes
W <- matrix(nrow = 500, ncol = p, data = w)

WW <- t(W) %*% W
try(solve(WW))

## Error in solve.default(WW) :
##   Lapack routine dgesv: system is exactly singular: U[5,5] = 0
is.singular.matrix(WW)

## [1] TRUE
```

## Dealing with $p \gg n$

- ▶ **Problem:** Variance increases with  $p$ ;  $\mathbf{W}'\mathbf{W}$  is singular
- ▶ **Solution:** Treat markers as random; shrink estimates towards 0
  - ▶ Penalization methods
  - ▶ Bayesian methods

## Penalized methods

- ▶ **Goal is to balance goodness-of-fit and model complexity**
  - ▶ Goodness-of-fit: minimize the residual sum of squares

$$\sum_i (y_i - \mu - \sum_{j=1}^p a_j w_{ij})^2$$

- ▶ Model complexity: function of model unknowns

$$J(a)$$

## Penalized methods

- ▶ How do we obtain estimates for marker effects?
- ▶ Solve the optimization function

$$(\hat{\mu}, \hat{a}) = \operatorname{argmin} \left\{ \sum_i (y_i - \mu - \sum_{j=1}^p a_j w_{ij})^2 + \lambda J(a) \right\}$$

-  $\lambda$  is regularization parameter;  $J(a)$  is penalty function

- ▶ Minimizing the **penalized** sum of squares

## Ridge regression BLUP

$$(\hat{\mu}, \hat{a}) = \operatorname{argmin} \left\{ \sum_i (y_i - \mu - \sum_{j=1}^p a_j w_{ij})^2 + \frac{\sigma_e^2}{\sigma_a^2} \sum_{j=1}^p a_j^2 \right\}$$

- ▶ Penalty function:  $J(a) = \sum_{j=1}^p a_j^2$ 
  - ▶ L2 Norm
- ▶ Regularization parameter:  $\lambda = \frac{\sigma_e^2}{\sigma_a^2} \operatorname{Var}(a) = \sigma_a^2 \mathbf{I}$

## Ridge regression BLUP

$$(\hat{\mu}, \hat{a}) = \operatorname{argmin} \left\{ \sum_i (y_i - \mu - \sum_{j=1}^p a_j w_{ij})^2 + \frac{\sigma_e^2}{\sigma_a^2} \sum_{j=1}^p a_j^2 \right\}$$

$$\hat{a} = [\mathbf{W}'\mathbf{W} + \lambda \mathbf{I}]^{-1} \mathbf{W}'\mathbf{y}$$

- ▶ Shrinks  $\hat{a}$  towards zero (increases bias, but reduces variance)

2019-06-25

# Bayesian whole-genome regression (Bayesian alphabet) for GWAS

## └ Ridge regression BLUP

$$(\hat{\mu}, \hat{\alpha}) = \operatorname{argmin}_{(\mu, \alpha)} \left\{ \sum_i (y_i - \mu - \sum_{j=1}^p \alpha_j w_{ij})^2 + \frac{\sigma^2}{\lambda} \sum_{j=1}^p \alpha_j^2 \right\}$$
$$\hat{\alpha} = [\mathbf{W}'\mathbf{W} + \lambda I]^{-1} \mathbf{W}'\mathbf{y}$$

- Shrinks  $\hat{\alpha}$  towards zero (increases bias, but reduces variance)

How does this compare to the optimization problem for OLS?? What happens when lambda is near 0? What about when sigma e » sigma g?

## Ridge regression BLUP: Summary

$$y = \mu + \mathbf{W}\mathbf{a} + e$$

$$\mathbf{a} \sim N(0, \mathbf{I}\sigma_a^2) ; e \sim N(0, \mathbf{I}\sigma_e^2)$$

$$\hat{\mathbf{a}} = [\mathbf{W}'\mathbf{W} + \lambda\mathbf{I}]^{-1}\mathbf{W}'\mathbf{y}$$

- ▶ **All markers** have a small, non-zero contribution to the phenotype (infinitesimal model)

2019-06-25

# Bayesian whole-genome regression (Bayesian alphabet) for GWAS

## └ Ridge regression BLUP: Summary

$$\begin{aligned}y &= \mu + \mathbf{W}\alpha + \epsilon \\ \alpha &\sim N(\mathbf{0}, \mathbf{I}\sigma^2_\alpha) ; \epsilon \sim N(\mathbf{0}, \mathbf{I}\sigma^2_\epsilon)\end{aligned}$$

$$\hat{\alpha} = [\mathbf{W}'\mathbf{W} + \lambda\mathbf{I}]^{-1}\mathbf{W}'y$$

- All markers have a small, non-zero contribution to the phenotype (infinitesimal model)

Other approaches (e.g. LASSO) allow marker effect to be zero.?? What do you think about this approach? Are our assumptions reasonable?

# Ridge regression using rrBLUP

Kinship (G) matrix  
For G-BLUP only!

```
```{r, echo = T, eval = F}
rrblup.mod <- mixed.solve(y, Z = NULL, K = NULL, X = NULL, method="REML", SE=F)
````
```

Phenotypes

```
[1] 0.3128947 0.3984038 0.3995326 0.3202826 0.5515313 0.3495400
```

Random effects  
(SNPs)

```
> head(t(genos)[,1:5])
mlid0000002064 mlid0000004235 mlid0000004358
X006dfe9b.0      1      1      1
X02cc7c6d.0      1      1     -1
X02d095ba.0      1      1      1
X07dac217.0      1      1      1
X07f246bb.0      1      1      1
X08de34ee.0      1      1     -1
```

# Thinking like a Bayesian

- ▶ Frequentist: “What is the best estimate for model unknowns given the data?”
- ▶ Bayesian: “What is the posterior density of the model unknowns given the data and hyperparameters?”

$$p(\mu, \mathbf{a}, \sigma^2 | \mathbf{y}, \omega)$$

- ▶ This is proportional to the conditional density of the data given the unknowns and the joint prior density of model unknowns

$$p(\mu, \mathbf{a}, \sigma^2 | \mathbf{y}, \omega) \propto p(\mathbf{y} | \mu, \mathbf{a}, \sigma^2) p(\mu, \mathbf{a}, \sigma^2 | \omega)$$

## Bayesian Ridge regression

- ▶ Optimization function for RR:

$$(\hat{\mu}, \hat{a}) = \operatorname{argmin} \left\{ \sum_i (y_i - \mu - \sum_{j=1}^p a_j w_{ij})^2 + \frac{\sigma_e^2}{\sigma_a^2} \sum_{j=1}^p a_j^2 \right\}$$

- ▶ After some fancy transformations:

$$(\hat{\mu}, \hat{a}) = \operatorname{argmin} \left\{ \prod_i^n \exp \left( \frac{(y_i - \mu - \sum_{j=1}^p a_j w_{ij})^2}{2\sigma^2} \right) \right\} \left\{ \prod_{j=1}^p \exp \left( -\frac{a_j^2}{2\sigma^2} \right) \right\}$$

## Bayesian Ridge Regression (BRR)

$$(\hat{\mu}, \hat{a}) = \operatorname{argmin} \left\{ \prod_i^n \exp \left( \frac{(y_i - \mu - \sum_{j=1}^p a_j x_{ij})^2}{2\sigma^2} \right) \right\} \left\{ \prod_{j=1}^p \exp \left( -\frac{a_j^2}{2\sigma^2} \right) \right\}$$

- ▶ Probability density function for normal distribution  
 $p(x|a, b) = \frac{1}{\sqrt{2\pi b}} \exp \left( -\frac{(x-a)^2}{2b} \right)$
- ▶ The mode of the posterior of the Bayesian model is equivalent to the RR solution
  - ▶ Bayesian Ridge Regression = Bayesian WGR with Gaussian prior

# BRR using the BGLR package

Kinship (G) matrix  
For G-BLUP only!

```
```{r, echo = T, eval = F}
rrblup.mod <- mixed.solve(y, Z = NULL, K = NULL, X = NULL, method="REML", SE=F)
````
```

Phenotypes

```
[1] 0.3128947 0.3984038 0.3995326 0.3202826 0.5515313 0.3495400
```

Random effects (SNPs)

```
> head(t(genos)[,1:5])
mlid0000002064 mlid0000004235 mlid0000004358
X006dfe9b.0      1      1      1
X02cc7c6d.0      1      1      -1
X02d095ba.0      1      1      1
X07dac217.0      1      1      1
X07f246bb.0      1      1      1
X08de34ee.0      1      1      -1
```

# BRR using the BGLR package

Incidence matrix  
for fixed effects

|      | [,1] | [,2] | [,3] | [,4] |
|------|------|------|------|------|
| [1,] | 1    | 0    | 0    | 0    |
| [2,] | 0    | 1    | 0    | 0    |
| [3,] | 0    | 0    | 1    | 0    |
| [4,] | 0    | 0    | 0    | 1    |
| [5,] | 1    | 0    | 0    | 0    |
| [6,] | 0    | 1    | 0    | 0    |

Prior

```
ETA <- list(list(X = Xincid, model = "FIXED"),
            list(X = genos, model = "BRR"))
```

Random effects  
(SNPs)

|             | mlid0000002064 | mlid0000004235 | mlid0000004358 | mlid0000004878 | mlid0000005926 |
|-------------|----------------|----------------|----------------|----------------|----------------|
| X006df9b.0  | 2              | 2              | 2              | 2              | 2              |
| X02cc7c6d.0 | 2              | 2              | 0              | 2              | 2              |
| X02d095ba.0 | 2              | 2              | 2              | 2              | 2              |
| X07dac217.0 | 2              | 2              | 2              | 2              | 2              |
| X07f246bb.0 | 2              | 2              | 2              | 0              | 2              |
| X08de34ee.0 | 2              | 2              | 0              | 2              | 2              |

## Straying from the infinitesimal genetic architecture

- ▶ Will all traits follow an infinitesimal model?
  - ▶ infinitesimal model = QTL effects are normally distributed
- ▶ How can we modify the Bayesian approach to fit different genetic architectures?

## Straying from the infinitesimal genetic architecture

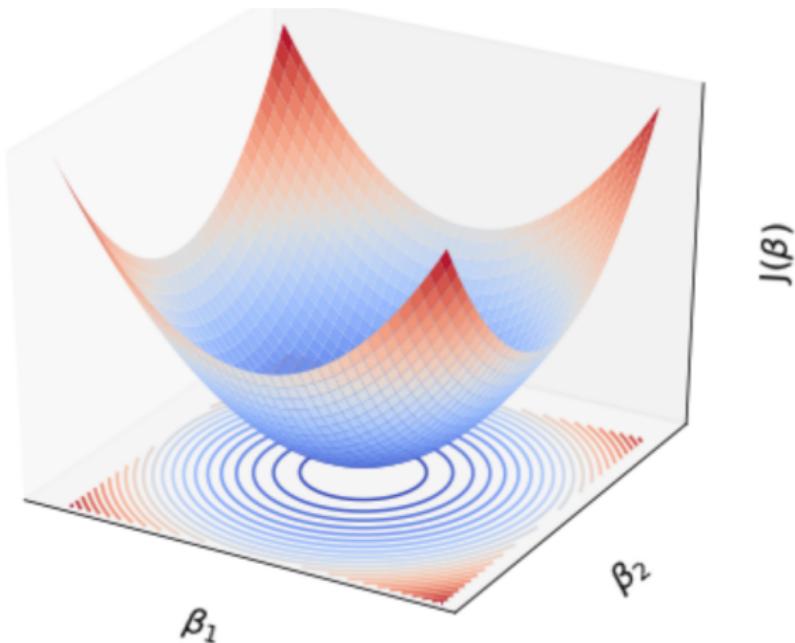
- ▶ Will all traits follow an infinitesimal model? **probably not**
  - ▶ infinitesimal model = QTL effects are normally distributed
- ▶ How can we modify the Bayesian approach to fit different genetic architectures? **choose a different prior**

# LASSO

- ▶ Least Absolute Shrinkage and Selection Operator
  - ▶ **Variable selection:** Some marker effects will be 0, others non-0
  - ▶ Like RR, LASSO can be interpreted from both a frequentist and Bayesian perspective
- ▶ Frequentist LASSO (L1 norm)

$$(\hat{\mu}, \hat{\mathbf{a}}) = \operatorname{argmin} \left\{ \sum_i (y_i - \mu - \sum_{j=1}^p w_{ij} a_j)^2 + \frac{\sigma_e^2}{\sigma_a^2} \sum_{j=1}^p |a_j| \right\}$$

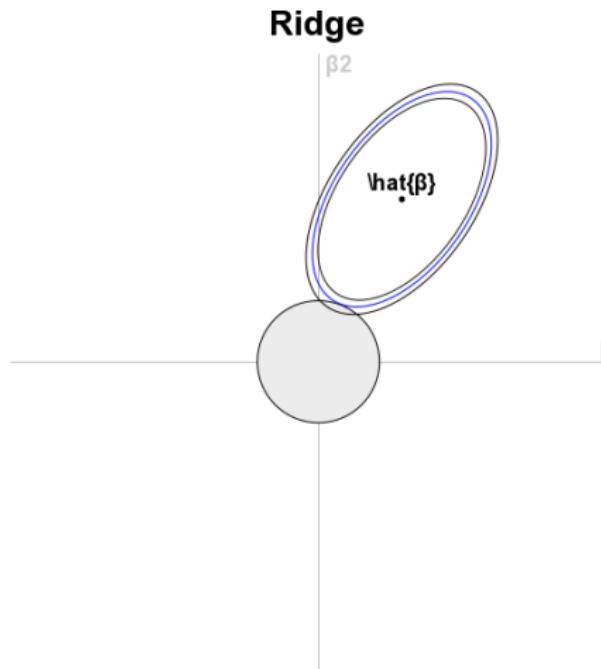
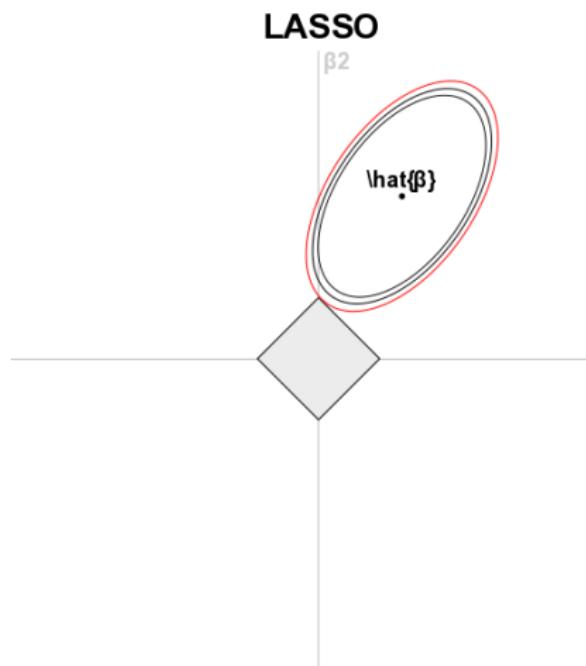
## Variable selection with LASSO



- ▶  $x$  and  $y$ -axis are the coefficients for the predictors;  $z$ -axis is the error
  - ▶ OLS will find solutions to minimize the error

## Variable selection with LASSO

- ▶ Goal: find first point where the elliptical contours intersect the constraint region ( $\sum_{j=1}^p |a_j|^q$ )



- ▶ Blue ellipse is the RR solution, red is LASSO solution

## Bayesian LASSO

$$(\hat{\mu}, \hat{a}) = \operatorname{argmin} \left\{ \sum_i (y_i - \mu - \sum_{j=1}^p a_j w_{ij})^2 + \frac{\sigma_e^2}{\sigma_a^2} \sum_{j=1}^p |a_j| \right\}$$

$$(\hat{\mu}, \hat{a}) = \operatorname{argmin} \left\{ \prod_i^n \exp \left( \frac{(y_i - \mu - \sum_{j=1}^p a_j w_{ij})^2}{2\sigma^2} \right) \right\} \left\{ \prod_{j=1}^p \exp \left( -\frac{\lambda |a_j|}{2\sigma^2} \right) \right\}$$

- The posterior is the same as RR, but the **prior is different**

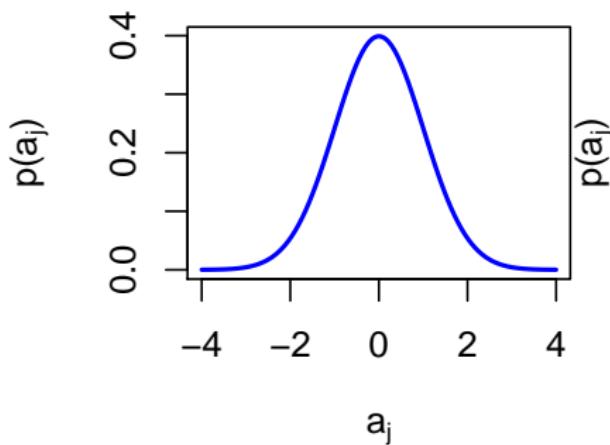
# Bayesian LASSO

$$(\hat{\mu}, \hat{a}) = \operatorname{argmin} \left\{ \prod_i^n \exp \left( \frac{(y_i - \mu - \sum_{j=1}^p x_{ij} a_j)^2}{2\sigma^2} \right) \right\} \left\{ \prod_{j=1}^p \exp \left( -\frac{\lambda |a_j|}{2\sigma^2} \right) \right\}$$

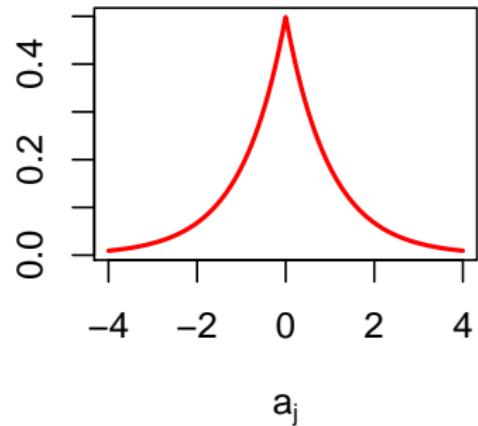
- ▶ Probability density function for Laplace distribution:  
 $f(x|a, b) = \frac{1}{2b} \exp\left(-\frac{|x-a|}{b}\right)$

## Priors for Bayesian ridge and LASSO

$N(0, 1)$



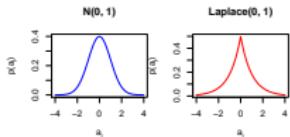
$Laplace(0, 1)$



2019-06-25

## Bayesian whole-genome regression (Bayesian alphabet) for GWAS

### └ Priors for Bayesian ridge and LASSO



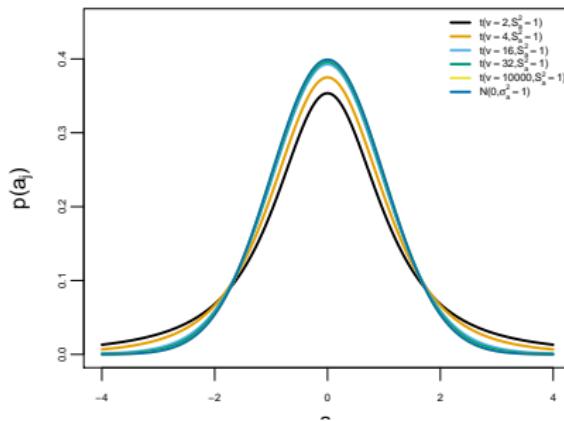
Notice the differences between the two densities. The Laplace distribution has a higher mass at zero and thicker tails compared to the Gaussian density. What does this mean? Small effect markers are set to zero, while markers with larger effects are shrunk less.

## Different Bayesian flavors

- ▶ Bayesian Ridge Regression
- ▶ Bayesian LASSO
- ▶ BayesA
- ▶ BayesB
- ▶ BayesC
- ▶ BayesC $\pi$

## BayesA: scaled $t$

- ▶ Bayes A: Each marker has a marker-specific variance  
 $a_j \sim N(0, \sigma_{a_j}^2)$
- ▶ Prior: Infinite mixture of normal distributions = scaled  $t$   
( $t(a_j | v, S_a^2)$ )
  - ▶ Scale ( $S_a^2$ ); degrees of freedom ( $v$ )

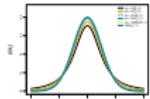


# Bayesian whole-genome regression (Bayesian alphabet) for GWAS

## └ BayesA: scaled $t$

### BayesA: scaled $t$

- ▶ Bayes A: Each marker has a marker-specific variance  
 $a_j \sim N(0, s_{\alpha_j}^2)$
- ▶ Prior: Infinite mixture of normal distributions = scaled  $t$   
( $t(a_j | v, S_a^2)$ )
  - ▶ Scale ( $S_a^2$ ), degrees of freedom ( $v$ )



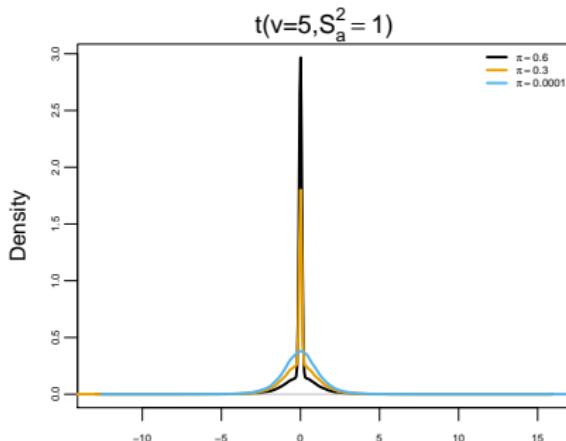
How does shrinkage with BayesA compare to RR?

## BayesA vs ridge regression

- ▶ Small-effect loci: Shrunk closer to 0 with BayesA compared to RR
- ▶ Large-effect loci: Less shrinkage with BayesA compared to RR
- ▶ How is shrinkage affected by degrees of freedom ( $v$ )?

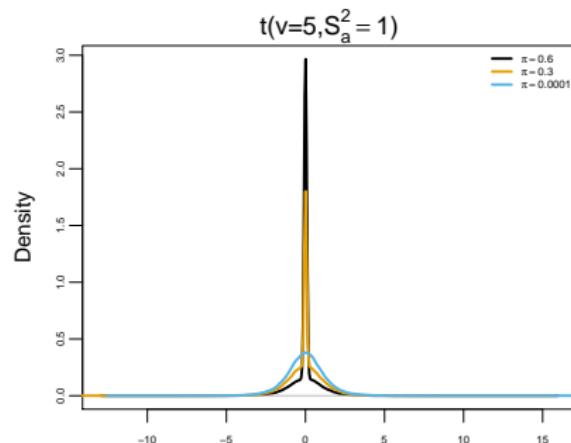
## BayesB: spike-slab density

- ▶ Like BayesA, with BayesB each marker has a marker-specific variance
- ▶ Prior: Spike-slab
  - ▶ Spike: marker effect will have point mass at 0 with a probability  $\pi$
  - ▶ Slab: marker effect will have  $t$  distribution with scale ( $S_a^2$ ) and degrees of freedom ( $v$ ), and **a probability**  $1 - \pi$



# BayesB vs Bayes A

- ▶ How does shrinkage compare between BayesA and BayesB for small-effect loci?
  - ▶ Assume  $\pi = 0.3$



## Bayes C and BayesC $\pi$

- ▶ BayesC:
  - ▶ Nearly the same as BayesB, however the prior is slightly different
  - ▶ Prior: Spike-slab
    - ▶ Spike: marker effect will have point mass at 0 with a probability  $\pi$
    - ▶ Slab: marker effect will follow a normal distribution  $(N(0, \sigma_a^2 | v, S_a^2))$  and a **probability**  $1 - \pi$
  - ▶ Assumes  $\pi$  is known
- ▶ BayesC $\pi$ :  $\pi$  is unknown
  - ▶ Uniform (flat, uninformative) prior is assigned to  $\pi$

## Summary of Bayesian WGR methods

- ▶ **BRR:**  $a$ 's assigned  $N(0, \sigma_a^2)$ ; variance parameter ( $\sigma_a^2$ ) assigned  $\chi^{-2}(\sigma_a^2 | df_a, S_a)$
- ▶ **BL:**  $a$ 's assigned  $N(0, \tau_{jk}^2 \sigma_e^2)$ ; marker-specific variance parameter ( $\tau_j^2$ ) assigned  $DE(\tau_{jk}^2 | \frac{\lambda^2}{2})$
- ▶ **BayesA:**  $a$ 's assigned  $N(0, \sigma_{aj}^2)$ ; each marker-specific variance ( $\sigma_{aj}^2$ ) is assigned  $\chi^{-2}(\sigma_{aj}^2 | df_a, S_a)$  which is the same as a scaled  $t$  dist.
- ▶ **BayesB:**  $a$ 's assigned 0 with probability  $\pi$  and  $N(0, \sigma_{aj}^2)$  with probability  $(1 - \pi)$ ; each marker-specific variance ( $\sigma_{aj}^2$ ) is assigned  $\chi^{-2}(\sigma_{aj}^2 | df_a, S_a)$  which is the same as a scaled  $t$  dist.
- ▶ **BayesC:**  $a$ 's assigned 0 with probability  $\pi$  and  $N(0, \sigma_a^2 | v, S_a^2)$  with probability  $(\pi)$ .
  - ▶ **BayesC $\pi$ :** BayesC with a flat prior on  $\pi$

# Inference on Bayesian models

What is the genetic architecture for trait X?

1. Define the model: Prior knowledge of unknown parameters, data generating process (likelihood)
  - ▶  $p(\mu, \alpha, \sigma^2 | y, \omega) \propto \prod_{i=1}^n N(y_i | \mu + \sum_{j=1}^p w_{ij} \alpha_j, \sigma^2) \prod j = 1^p p(\alpha_j | \omega) p(\sigma^2)$
2. Given some data estimate the unknown parameters via Markov chain Monte Carlo method
  - ▶ Monte Carlo: Random sampling
  - ▶ Markov Chain: Sequence where value for current iteration depends on previous iteration

## Gibbs sampling (very brief)

0. Set initial values for all unknown parameters
1. For iteration  $i$  in 1:nIter
  - i. For predictor  $p$  in 1:nPred, sample variable  $p$  conditional on all others
  - ii. Repeat i. until  $p == nPred$
2. Repeat 1.

Visualization of simple distribution

## P-values for marker effects

- ▶ For frequentist approaches calculating  $p$ -values is simple, for Bayesian... not so much
- ▶ p-values for ridge regression
  1. Define  $t$ -statistic:  $t = \frac{\hat{a}}{\sqrt{Var(\hat{a})}}$
  2. Get  $p$ -value:  $p-value = 2(1 - \Phi(|t|))$ ;  $\Phi$  is the CDF for the Normal dist.

# P-values for RR-BLUP

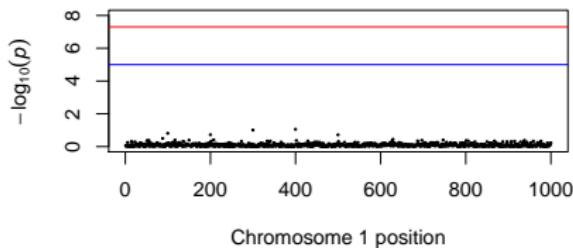
## ► In rrBLUP

```
foo <- mixed.solve(y = y, Z = t(geno[4:ncol(geno)]),  
                    K = NULL, X = NULL, SE = T)  
str(foo)  
  
## List of 7  
## $ Vu      : num 0.00594  
## $ Ve      : num 6.09  
## $ beta    : num [1(1d)] -0.227  
## $ beta.SE: num [1(1d)] 0.241  
## $ u       : num [1:1000(1d)] -0.01349 0.00365 0.00734 -0.01832 -0.01...  
## $ u.SE    : num [1:1000(1d)] 0.0734 0.0734 0.0733 0.0734 0.0734 ...  
## $ LL      : num -528
```

# P-values for RR-BLUP

## ► In rrBLUP

```
foo <- mixed.solve(y = y, Z = t(geno[4:ncol(geno)]),  
                    K = NULL, X = NULL, SE = T)  
  
SNPe_ad <- foo$u / foo$u.SE  
pvals <- 2*(1-pnorm(abs(SNPe_ad)))
```



## Bayesian inference

- ▶ Frequentist  $p$ -values: the strength of association between  $y$  and  $x$
- ▶ Bayesian model frequency: proportion of samples in which  $a$  has a non-zero effect
  - ▶ BayesB
  - ▶ BayesC and BayesC $\pi$

Questions??