BAYESIAN HIERARCHICAL REGRESSION MODEL TO DETECT QUANTITATIVE TRAIT LOCI

Haikun Bao

A Thesis Submitted to the
University of North Carolina Wilmington in Partial Fulfillment
Of the Requirements for the Degree of
Master of Arts

Department of Mathematics and Statistics

University of North Carolina Wilmington

2006

Approved by

Advisory Committee

____________________ ____________________

____
Chair

Accepted by

____
Dean, Graduate School

TABLE OF CONTENTS

ABSTRACT

Detecting genetic loci responsible for variation in quantitative traits is a problem of great importance to biologists. The location on a genetic map responsible for a quantitative trait is referred to as Quantitative Trait Loci, or QTL. This thesis uses a Bayesian Hierarchical Regression model which incorporates variability both within and between lines to detect the QTL. This method is applied to a simulated data set using the line information from Bay-0 × Shahdara population to find the activation probability of each genetic segment via the Gibbs sampler and Monte Carlo integration techniques. Using the activation probability, which indicates the influence of each segment within all the models, the QTL is detected. The results show that it is an effective way to detect QTL.

ACKNOWLEDGMENTS

First and most of all, I would like to thank my advisor, Dr. Susan Simmons, for her great insight, guidance, patience, encouragement and being there for me every step of the way. This could not have been possible without her.

I would also like to thank Dr. Boone, for assisting me in my understanding of theory relevant to this thesis. I also like to express my sincere gratitude to all the faculty members and fellow students in the Department of Mathematics and Statistics for sharing their love and knowledge of mathematics and statistics with me.

LIST OF TABLES

LIST OF FIGURES

INTRODUCTION

The identification of genetic loci responsible for variation in traits that are quantitative in nature is a problem of great importance to biologists. Quantitative Trait Loci (QTL) analysis is the search for the location or loci on a genetic map responsible for controlling a quantitative trait. The QTLs help researchers understand the biochemical basis of these traits, and their evolution in populations over time. Moreover, knowledge of these loci may aid in the design of future experiments to manipulate these traits [1].

A genetic map shows the location of genetic markers along the chromosome and its relative distance. One of the main goals of QTL analysis is to find the locations on the genetic map most responsible for differences in a quantitative trait. Examples of a quantitative trait are yield of a crop, angle opening of a flowering plant, height of a plant, etc.

One of the earliest QTL methods was to perform an Analysis of Variance (ANOVA), or one marker at a time analysis[1]. Since then, many more sophisticated algorithms have evolved. Some of the most recent methods include Bayesian regression, model selection search, composite interval mapping, multiple interval mapping, and even some hierarchical modeling. There are a number of softwares available that perform QTL analysis such as QTL Cartographer [2], BQTL [3], and RQTL [4]. However most of the software packages available require only one observation per genotype(or line).

QTL experiments involving plants will often produce multiple observations per genotype or line. Although the observations in one line can be considered independent of each other they are in reality "clones" because they have identical genetic composition. To utilize existing software, most plant biologists will take the average value (or the median value) of the quantitative trait within each line to perform a

QTL analysis.

In a plant QTL experiment, the number of clones within line $i(i = 1, ..., L)$ is n_i. The n_i clones within each line have the same marker information on their genetic maps. As mentioned previously, when plant biologist perform QTL analysis, the n_i plants in one line are averaged to obtain one value. However, by doing this, important information is lost. For example, if we have two lines: Line 1 has the following information on its clones: 30, 40, 40, 50. Line 2 has the following information on its clones: 0, 20, 100. By using the average as the measured trait value within each line, Line 1 and Line 2 look identical with a mean of 40. However, Line 1 provides more information regarding the quantitative trait because it has smaller variability, where Line 2 provides less information. This extra level of variability should be included in the model. This thesis will address the problem of incorporating the extra level of variability via a Bayesian hierarchical regression model and will apply this method to a simulated data set.

BACKGROUND

Bayesian Data Analysis

Bayesian data analysis is based on probability models for observed quantities and those in which we would like to make inferences. Probability is used to quantify uncertainty in inference within Bayesian methods. The conclusions about unknown parameters, and unobserved data are made in terms of probability statements [5].

Bayesian methods provide results which are, at times, easier to interpret and understand than frequentist methods. One example is the Bayesian probability interval for an unknown quantity of interest. This interval has the interpretation that "the probability that the unknown random quantity is contained in the interval is $(1-\alpha) * 100\%$". Whereas the frequentist interval can only be interpreted with respect to the "confidence" that the unknown quantity lies in the interval.

In dealing with very complex problems, the Bayesian framework has great advantages for its flexibility and generality. In a Bayesian analysis, it is easy to incorporate new information into an already existing model. Another advantage of Bayesian method is the ability to create multilayered probability specifications.

Bayesian data analysis can be divided into the following three steps:
1) The full probability model involving all the observable and unobservable quantities in a problem must be specified. This model is developed using knowledge about the underlying scientific problem and the data collection process. The Bayesian data analysis is based on this step. 2) Incorporating information obtained from the observed data. The posterior distribution of quantities of interest is derived by conditioning on the observed data. Here the posterior distribution is a conditional probability function. The posterior distribution of parameters can be used to construct confidence intervals, test hypotheses or other inferential procedures. 3) Finally, the fit of the model and the implication of the resulting posterior distribu-

tion need to be evaluated. It is important to know whether or not the model fits the data well, if the substantive conclusions are reasonable, and how sensitive the results are to the modeling assumptions in Step 1 [5]. This thesis will address steps 1 and 2 of the proposed model; however, step 3 will be left for future research.

Bayesian Inference

In a Bayesian setting, parameters are unknown random quantities and therefore have a distribution. In order to make probability statements about an unknown parameter θ given the data y, we begin with a model providing a joint probability distribution for θ and y.

$$p(\theta, y) = p(\theta)p(y \mid \theta) \tag{1}$$

The quantity $p(\theta)$, known as the prior distribution of θ, is assumed to be constructed from prior knowledge and expert advice. The quantity $p(y \mid \theta)$ is the sampling distribution of the observed data. Using (1) and the definition of conditional probability, the posterior density is:

$$p(\theta \mid y) = \frac{p(\theta, y)}{p(y)} = \frac{p(\theta)p(y \mid \theta)}{p(y)} \tag{2}$$

The $p(y \mid \theta)$ is called the likelihood function when it is regard as a function of θ, for observed y. The data y effect the posterior distribution $p(\theta \mid y)$ only through the likelihood function with a chosen probability model. The ratio of the posterior density $p(\theta \mid y)$ evaluated at the points θ_1 and θ_2 is called posterior odds.

$$\frac{p(\theta_1 \mid y)}{p(\theta_2 \mid y)} = \frac{p(\theta_1)p(\theta_1 \mid y)/p(y)}{p(\theta_2)p(\theta_2 \mid y)/p(y)} = \frac{p(\theta_1)}{p(\theta_2)} \cdot \frac{p(y \mid \theta_1)}{p(y \mid \theta_2)} \tag{3}$$

From (3), the posterior odds are equal to the prior odds multiplied by the likelihood ratio.

Summarizing Inferences by Simulation

Simulation plays an important role in applied Bayesian analysis. The samples can be generated from a probability distribution, even when the density function cannot be explicitly integrated.

To simulate the posterior distribution of unknown parameter θ, we obtain samples from discrete and continuous prior distributions by using the inverse cumulative distribution function or some other technique for obtaining random samples from $p(\theta)$.

The posterior distribution in (2) contains three probability distributions: $p(\theta)$, $p(y|\theta)$ and $p(y)$. The marginal distribution of y, $p(y)$, is a normalizing constant with respect to θ for the posterior distribution of θ [5], so we can write

$$p(\theta|y) \propto p(y|\theta)p(\theta) \tag{4}$$

We can use (4) to simulate the posterior distribution of θ by three steps: 1) Obtain a random draw θ_i from $p(\theta)$. 2) Using θ_i from step 1 to obtain a random draw from $p(y|\theta_i)$. Step 1 and 2 create a random draw from $p(y,\theta)$ and can be repeated many times to get a random sample from this joint distribution. We can further use this information to get an approximate posterior distribution.

With simulating draws from the posterior distribution of θ, we can estimate the posterior probabilities of any quantity of interest. For instance, we can compute posterior probability intervals, $p(a < \theta < b)$, for given a and b by the proportion in which this event is true over the simulation.

Gibbs Sampler

Markov Chain Monte Carlo simulation is a general method to get draws from the posterior distribution. It draws values of the model parameters from approximate distributions and corrects those draws to better approximate the target posterior distribution. The chain needs initial starting values, and then sequentially draws and updates parameters from the approximate distributions. The approximate distributions are improved at each step in the simulation and convergent to the target distribution. [5]

Gibbs sampler, a particular Markov Chain Monte Carlo algorithm, is very useful in the multidimensional problem. In this method, the parameter vector is divided into l components, $\theta = (\theta_1 ... \theta_l)$. Each iteration draws a subset of the parameters conditional on the value of all the others. For example, an ordering of the l subvectors of θ is chosen and at each iteration t, the subset θ_j^t is sampled from the conditional distribution given all the other subsets of θ, i.e $p(\theta_j|\theta_1^t, ..., \theta_{j-1}^t, \theta_{j+1}^{t-1}, ..., \theta_l^{t-1}, y)$, where θ_{j+1}^{t-1} is the sampled value of θ_{j+1} in $t-1$ iteration.

For example, assume (y_1, y_2) are from a bivariate normal distribution with unknown mean (θ_1, θ_2) and known covariance matrix $(\begin{matrix} 1 & \rho \\ \rho & 1 \end{matrix})$ [5], where ρ is fixed covariance of y_1 and y_2. With uniform prior distribution of θ, we have the following posterior distribution

$$\begin{pmatrix} \theta_1 \\ \theta_2 \end{pmatrix} |y \sim N\left(\begin{pmatrix} y_1 \\ y_2 \end{pmatrix}, \begin{pmatrix} 1 & \rho \\ \rho & 1 \end{pmatrix}\right) \tag{5}$$

The full conditional posterior distribution are

$$\begin{aligned} \theta_1|\theta_2, y &\sim N(y_1 + \rho(\theta_2 - y_2), 1 - \rho^2) \\ \theta_2|\theta_1, y &\sim N(y_2 + \rho(\theta_1 - y_1), 1 - \rho^2) \end{aligned} \tag{6}$$

We can obtain a sample from the multivariate posterior distribution (5) by choosing initial values of θ_1 and θ_2 to start the chain. The value of θ_2 is used to draw a random value of θ_1 from $p(\theta_1|\theta_2, y)$ in (6). This updated value of θ_1 is then used to draw a random value of θ_2 from $p(\theta_2|\theta_1, y)$ in (6). This idea is continued for t iterations giving t draws from the multivariate posterior distribution (5).

Prior Distribution

The prior distribution should include all plausible values of the unknown parameter. For example, if the unknown parameter is a variance parameter, then the prior distribution should only allow positive values. If the sample size is large, the information about unknown parameter contained in the data will provide more information to the posterior distribution than any prior probability specification. However, if the sample size is small, the prior is extremely influential in the posterior distribution. We can chose reasonable prior distributions in terms of our information and knowledge, and attempt to use conjugate prior distributions whenever possible. Conjugate priors simplify results since the posterior can usually be put in analytic form. The property that the posterior distribution follows the same parametric form as the prior distribution is called conjugacy.

The following are some examples of conjugate prior distributions:

(1) If the data are obtained through a binomial experiment, the likelihood function is of the form $p(y \mid \theta) \propto \theta^y(1-\theta)^{n-y}$. The conjugate prior for this distribution is the Beta distribution. If we assign $p(\theta) \sim Beta(\alpha, \beta)$, then the posterior is of the

form

$$\begin{aligned}
p(\theta \mid y) &\propto \theta^y(1-\theta)^{n-y}\theta^{\alpha-1}(1-\theta)^{\beta-1} \\
&= \theta^{y+\alpha-1}(1-\theta)^{n-y+\beta-1} \\
&\sim Beta(\theta \mid \alpha+y, \beta+n-y)
\end{aligned}$$

(2) If the likelihood function of the data is assumed to be normal with unknown mean θ and known variance σ^2, then the likelihood function for a sample of independent and identically distributed observations $y = (y_1, ..., y_n)$ is:

$p(y \mid \theta) = (\frac{1}{\sqrt{2\pi}\sigma})^n \prod e^{-\frac{(y_i-\theta)^2}{2\sigma^2}} \propto exp[(-\frac{1}{2\sigma^2})\sum(y_i-\theta)^2]$

where the θ is the unknown mean of a normal distribution and variance σ^2 is known. In this situation, the conjugate prior distribution is of the form:

$p(\theta) \propto exp(-\frac{1}{2\tau_0^2}(\theta-\mu_0)^2)$ namely

$\theta \sim N(\mu_0, \tau_0^2)$

Then the posterior distribution is:

$$\begin{aligned}
p(\theta \mid y) &\propto exp\left\{-\frac{1}{2}\left[\frac{\sum(y_i-\theta)^2}{\sigma^2} + \frac{(\theta-\mu_0)^2}{\tau_0^2}\right]\right\} \\
&\propto exp\left[-\frac{1}{2\tau_1^2}(\theta-\mu_1)^2\right]
\end{aligned}$$

where $\mu_1 = (\frac{1}{\tau_0^2}\mu_0 + \frac{n}{\sigma^2}\bar{y})/(\frac{1}{\tau_0^2} + \frac{n}{\sigma^2})$, $\frac{1}{\tau_1^2} = \frac{1}{\tau_0^2} + \frac{n}{\sigma^2}$, and

$\bar{y}$ is the mean of sample y.

So the posterior distribution is normal : $\theta \mid y \sim N(\mu_1, \tau_1^2)$

Hierarchical Model

In many instances, data will follow a hierarchical structure in which various parameters are associated in a hierarchical fashion. For example, a study of a fitness program was introduced to i facilities, with participants at facility i having fitness measure θ_i. It is reasonable to assume that the estimates of the θ_i's, which represents a sample of facilities, should be related. This can be accomplished by assuming the θ_i's are a sample from a common population distribution [5].

In plant QTL experiment, the observed data y_{ij} with $i = 1, ..., L$ and $j = 1, ..., n_i$ can be used to estimate the distribution of θ_i, which is the underlying true mean of line i. It is natural to model this problem hierarchically. We can model the observable outcomes y_{ij} conditionally on certain parameters θ_i's, which are given a probabilistic specification in terms of further parameters, known as hyperparameters. Figure 1 illustrates the structure of the data in a plant QTL experiment. The data, y_{ij} are obtained from a distribution with mean θ_i, and θ_i depends on the hyperparameters of β and τ.

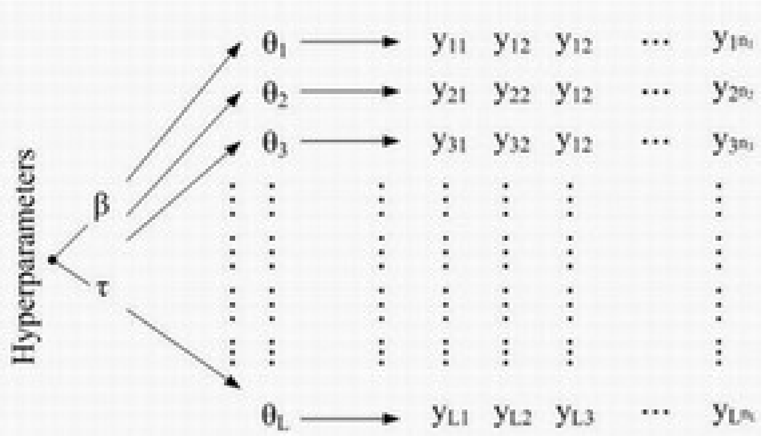

Figure 1: Structure of Hierarchical Model

THE QTL HIERARCHICAL MODEL

Data

The data for our model is assumed to be obtained through a design called Recombinant Inbred Line(RIL). RIL is a powerful design for detecting QTLs since in theory it has the largest variation between lines and the smallest variation within [6]. In this thesis, we use the line information from the Bay-0 × Shahdara(Bay × Sha) population to create a simulated QTL data set. The Bay-0 × Shahdara population was created by Olivier Loudet and Sylvain Chaillou between 1997 and 2000 at INRA Versailles [7]. Figure 2 illustrates the genetic map of the Bay × Sha RIL. The genetic map shows the location of genetic markers along five chromosomes and their relative distance. The marker information of the Bay × Sha population comprises the data matrix X. The X matrix is an L×M matrix where L is the number of lines and M is the number of markers. The simulated response matrix y is L×n matrix where L is the number of lines and n is the number of observations within each line. In this simulation, $n = 10$ within each line. We simulated a data set with one QTL located on marker 4 on the first chromosome(NGA248). The simulated response y_{ij} was created by first simulating the true θ_i in each line by (6)

$$\theta_i = 35 + m \times 3 + R_{normal} \tag{7}$$

Where $m = 0$, 1 or 2 depending on the value of the fourth marker of the first chromosome. The R_{normal} quantity is a random draw from the standard normal distribution. Using this information, we simulated 10 observations within each line by obtaining 10 random draws from a normal distribution with mean θ_i and standard deviation 0.2.

For the hierarchical model, we assume the observed data y_{ij} are normally

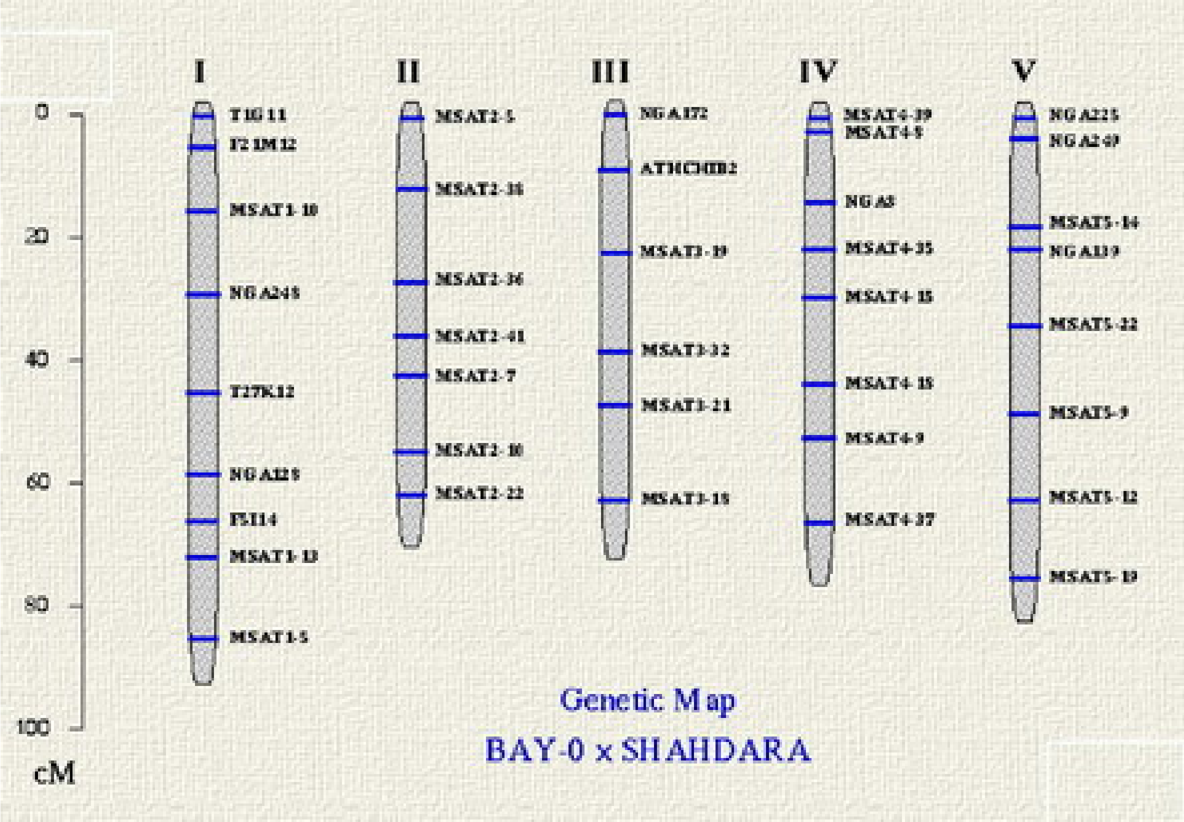

Figure 2: Genetic Map of Bay-0 × Shahdara population

distributed with mean θ_i and variance σ_i^2.

$$y_{ij} \mid \theta_i, \sigma_i^2 \sim N(\theta_i, \sigma_i^2)$$

The mean of the quantitative trait within each line, θ_i, is dependent on the marker information, so we model the θ_i using the regression model $Y = X\beta$ and assume the errors are independent with equal variance:

$$\theta \mid \beta, X, \tau^2 \sim N(X\beta, \tau^2 I)$$

where I is the L×L identity matrix.

We make the following assumptions regarding the prior distributions:

$$\begin{aligned} p(\sigma_i^2) &\sim Inv - \chi^2(\sigma_{0i}^2) \\ p(\beta_i) &\sim N(0, 100) \\ p(\tau^2) &\sim Inv - \chi^2(\tau_0^2) \end{aligned}$$

By setting $\sigma_{01}^2 = \sigma_{02}^2 = ... = \sigma_{0L}^2 = \tau_0^2 = 1$, the prior distribution of σ_i^2 and τ^2 have infinite means and variances [8]. The posterior distribution is:

$$\begin{aligned} p(\theta, \beta, \tau^2, \sigma^2 \mid y) &\propto \prod_i \prod_j \prod_k p(y|\theta, \beta, \tau^2, \sigma^2) p(\theta, \beta, \tau^2, \sigma^2) \\ &\propto \prod_i \prod_j \prod_k p(y|\theta, \sigma^2) p(\beta) p(\sigma^2) p(\tau^2) p(\theta|X\beta, \tau^2) \\ &\propto (\tau^{\tau_0+2+L} \prod_i (\sigma_i^{n_i+\sigma_{0i}+2}))^{-1} \cdot exp\left[-\sum_i \frac{1}{2\sigma_i^2} - \frac{1}{2\tau^2} - \frac{1}{200}\beta'\beta \right. \\ &\left. -\frac{1}{2\tau^2}(\theta - X\beta)'(\theta - X\beta) - \sum_i \sum_j \frac{1}{2\sigma_i^2}(y_{ij} - \theta_i)^2 \right] \end{aligned} \tag{8}$$

Where $i = 1, ..., L$ $j = 1, ..., 10$ and $k = 1, ..., M$

Posterior Distribution

We are interested in assessing how likely the observed data is given a chosen model. The models that we consider will keep all the θ_i's, σ_i^2's and τ; however, we are interested in understanding which β's are important in the model. Therefore, we will look at a number of models with and without various β's. The total number of models defined in this way is 2^M where M is the number of markers. We will denote the set of all possible models by Λ, and the K^{th} model by δ_K. The vector of unknown parameters for model K will be denoted by λ_K. Using this notation, the probability of the data given model K is:

$$p(D|\delta_K) = \int p(\theta \mid \beta_K, X_K, \tau^2)p(\tau^2)p(\sigma^2)p(\beta_K)p(y \mid \theta, \sigma^2)d\lambda_K \tag{9}$$

Here, $p(D|\delta_K)$ is the probability of the data given model K, where X_K is the genomic marker information matrix of model δ_K.

Gibbs Sampler for Hierarchical Model

To estimate the quantity (7), we need draws from the posterior distribution. We will use a particular Monte Carlo Markov chain algorithm, the Gibbs sampler, to obtain draws from the posterior distribution. In each iteration of the Gibbs sampler, we get a draw of parameters conditional on the values of all the other parameters. Thus there are four steps in the Gibbs sampler algorithm. At each iteration t, parameters $\tau^2, \theta, \beta, \sigma^2$ are sampled and updated conditional on the last values of the other parameters.

The initial starting points for the algorithm are important and we use the following information: $\beta^{(1)}$ comes from the β's of the regression model $y \sim x$; $\theta^{(1)}$ comes from the average value $\bar{y}$ of each line of y; $\sigma^{2^{(1)}}$ comes from the standard deviation of each line of y; and $\tau^{2^{(1)}}$ comes from the deviation of $\bar{y}$. The estimates of $\beta^{(1)}$, $\theta^{(1)}$, $\sigma^{2^{(1)}}$, and $\tau^{2^{(1)}}$ are the starting points for the Gibbs samplers to obtain random draws for β, θ, σ^2, and τ^2.

The random sample of all the parameters are obtained by generating random draws from each of the four full conditional distributions:

(1) To obtain random draws of τ^2's from the distribution of τ^2 conditional on

the other parameters $p(\tau^2|\theta,\beta,\sigma^2,y)$, we need

$$
\begin{aligned}
p(\tau^2|\theta,\beta,\sigma^2,y) &= \frac{p(\tau^2,\theta,\beta,\sigma^2|y)}{p(\theta,\beta,\sigma^2|y)} \\
&= \frac{p(\theta|X\beta,\tau^2)p(\tau^2)p(\sigma^2)p(\beta)p(y|\theta,\sigma^2)}{p(y|\theta,\sigma^2)p(\sigma^2)p(\beta)\int p(\theta|X\beta,\tau^2)p(\tau^2)d\tau^2} \\
&= \frac{p(\theta|X\beta,\tau^2)p(\tau^2)}{\int p(\theta|X\beta,\tau^2)p(\tau^2)d\tau^2} \\
&\propto \tau^{-(l+\tau_0^2+2)}exp\left\{-\frac{1}{2\tau^2}[(\theta-X\beta)'(\theta-X\beta)+1]\right\} \\
&\propto (\tau^2)^{-(\frac{l+\tau_0^2}{2}+1)}exp\left\{-\frac{[\,(\theta-X\beta)'(\theta-X\beta)+1]/2}{\tau^2}\right\}
\end{aligned}
$$

The conditional distribution of τ^2 is Inv-Gamma.

$$p(\tau^2|\theta,\beta,\sigma^2,y) \sim Inv-Gamma\left[\tfrac{l+\tau_0^2}{2},\tfrac{(\theta-X\beta)'(\theta-X\beta)+1}{2}\right]$$

(2) To obtain random draws of θ's from the distribution of θ conditional on the other parameters $p(\theta|\beta,\sigma^2,\tau^2,y)$, we need

$$
\begin{aligned}
p(\theta|\beta,\sigma^2,\tau^2,y) &= \frac{p(\tau^2,\theta,\beta,\sigma^2|y)}{p(\tau^2,\beta,\sigma^2|y)} \\
&= \frac{p(\theta|X\beta,\tau^2)p(y|\theta,\sigma^2)}{\int p(\theta|X\beta,\tau^2)p(y|\theta,\sigma^2)d\theta} \\
&\propto exp\left[-\frac{1}{2\tau^2}(\theta-X\beta)'(\theta-X\beta)-\sum_{i=1}^{L}\sum_{j=1}^{n_i}\frac{1}{2\sigma_i^2}(y_{ij}-\theta_i)^2\right] \\
&\propto exp\left\{\sum_{i=1}^{L}\left[-\frac{1}{2}\left(\frac{1}{\tau^2}+\frac{n_i}{\sigma_i^2}\right)\theta_i^2+\left(\frac{X_i\beta}{\tau^2}+\frac{C_i}{\sigma_i^2}\right)\theta_i\right]\right\} \\
&\propto exp\left\{\sum_{i=1}^{L}\frac{-1}{2\left(\frac{1}{\frac{1}{\tau^2}+\frac{n_i}{\sigma_i^2}}\right)}\left(\theta_i-\frac{\frac{X_i\beta}{\tau^2}+\frac{C_i}{\sigma_i^2}}{\frac{1}{\tau^2}+\frac{n_i}{\sigma_i^2}}\right)^2\right\}
\end{aligned}
$$

Where X_i is the i^{th} line of X, and $C_i = \sum_{j=1}^{n_i} y_{ij}$

The conditional distribution of θ_i is Normal.

$$p(\theta_i|\tau^2, \beta, \sigma^2, y) \sim N\left(\frac{\frac{X_i\beta}{\tau^2}+\frac{C_i}{\sigma_i^2}}{\frac{1}{\tau^2}+\frac{n_i}{\sigma_i^2}}, \frac{1}{\frac{1}{\tau^2}+\frac{n_i}{\sigma_i^2}}\right)$$

(3) To obtain random draws of β's from the distribution of β conditional on the other parameters $p(\beta|\theta, \sigma^2, \tau^2, y)$, we need

$$\begin{aligned}
p(\beta|\theta, \sigma^2, \tau^2, y) &= \frac{p(\theta|X\beta, \tau^2)p(\beta)}{\int p(\theta|X\beta, \tau^2)p(\beta)d\beta} \\
&\propto exp\left[\frac{-\beta'\beta}{200} - \frac{1}{2\tau^2}(\theta - X\beta)'(\theta - X\beta)\right] \\
&\propto exp\left\{-\frac{1}{2}\left[\beta'\left(\frac{I}{100}+\frac{X'X}{\tau^2}\right)\beta - \frac{2}{\tau^2}\theta' X\beta\right]\right\} \\
&\propto exp\left\{-\frac{1}{2}\left[\beta - \left(\frac{I}{100}+\frac{X'X}{\tau^2}\right)\frac{X'\theta}{\tau^2}\right]'\left(\frac{I}{100}+\frac{X'X}{\tau^2}\right)\right. \\
&\quad \left.\left[\beta - \left(\frac{I}{100}+\frac{X'X}{\tau^2}\right)\frac{X'\theta}{\tau^2}\right]\right\}
\end{aligned}$$

Where I is L×L identity matrix.

The conditional distribution of β is Normal.

$$p(\beta|\theta, \sigma^2, \tau^2, y) \sim N\left[\left(\tfrac{I}{100}+\tfrac{X'X}{\tau^2}\right)\tfrac{X'\theta}{\tau^2}, \left(\tfrac{I}{100}+\tfrac{X'X}{\tau^2}\right)^{-1}\right]$$

(4) To obtain random draws of σ^2's from the distribution of σ^2 conditional on the other parameters $p(\sigma^2|\tau^2, \theta, \beta, y)$, we need

$$\begin{aligned}
p(\sigma^2|\tau^2, \theta, \beta, y) &= \frac{p(y|\theta, \sigma^2)p(\sigma^2)}{\int p(y|\theta, \sigma^2)p(\sigma^2)d\sigma^2} \\
&\propto \prod^{L}(\sigma_i^2)^{-(\frac{\sigma_0^2}{2}+\frac{n_i}{2}+1)} exp\left\{-\left[\sum_{i=1}^{L}\frac{1}{2\sigma_i^2} + \sum_{i=1}^{L}\sum_{j=1}^{n_i}\frac{1}{2\sigma_i^2}(y_{ij}-\theta_i)^2\right]\right\} \\
&\propto \prod^{L}(\sigma_i^2)^{-(\frac{\sigma_0^2+n_i}{2}+1)} exp\left\{-\sum_{i=1}^{L}\left(\frac{1}{2\sigma_i^2}\right)\left[\sum_{j=1}^{n_i}(y_{ij}-\theta_i)^2 + 1\right]\right\}
\end{aligned}$$

The conditional distribution of σ_i^2 is Inv-Gamma.

$$p(\sigma_i^2|\tau^2, \theta, \beta, y) \sim Inv - Gamma\left[\frac{\sigma_0^2+n_i}{2}, \frac{\sum_{j=1}^{n_i}(y_{ij}-\theta_i)^2+1}{2}\right]$$

To diminish the effect of the starting distribution, we discard first $5,000$ from $100,000$ iterations for each parameter. We assume that the distribution of the simulated parameter values, for large enough iteration t, are close to the target distribution. Figure 3, Figure 4 and Figure 5 illustrate the sampled values from the Gibbs sampler for β_5,β_{16} and β_{32}.

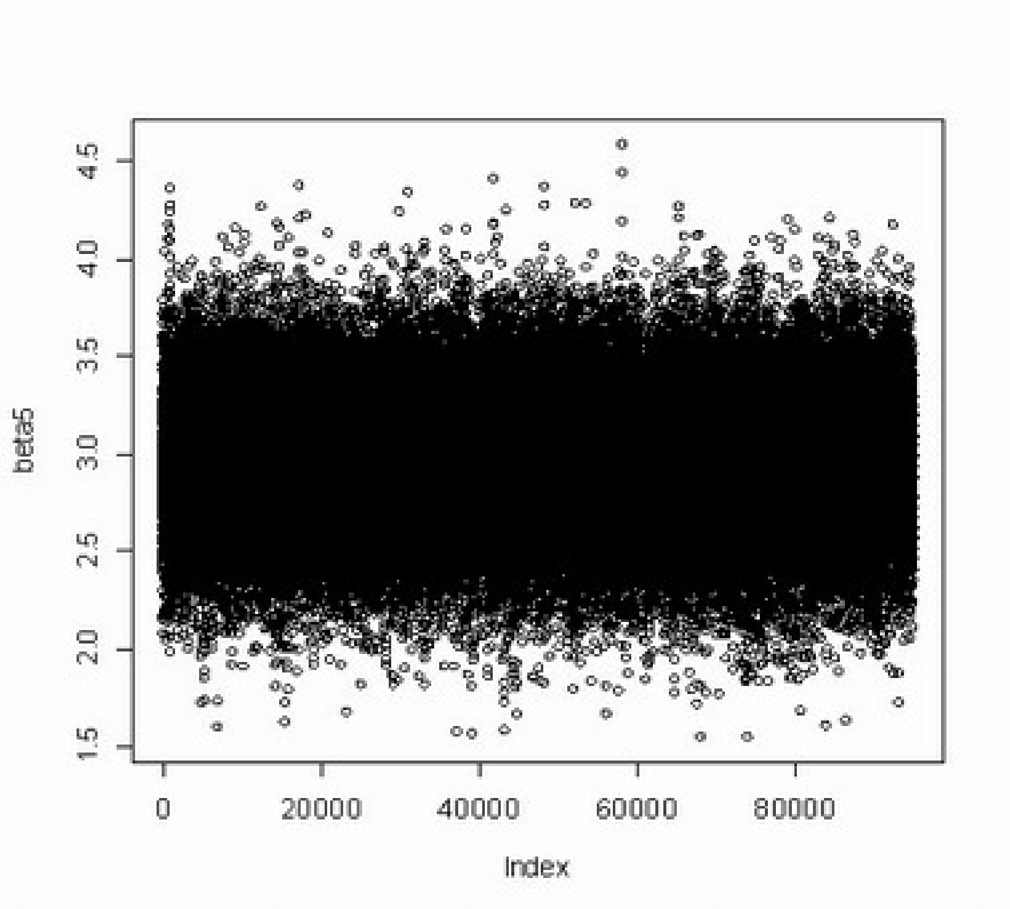

Figure 3: β_5 from Gibbs Sampler

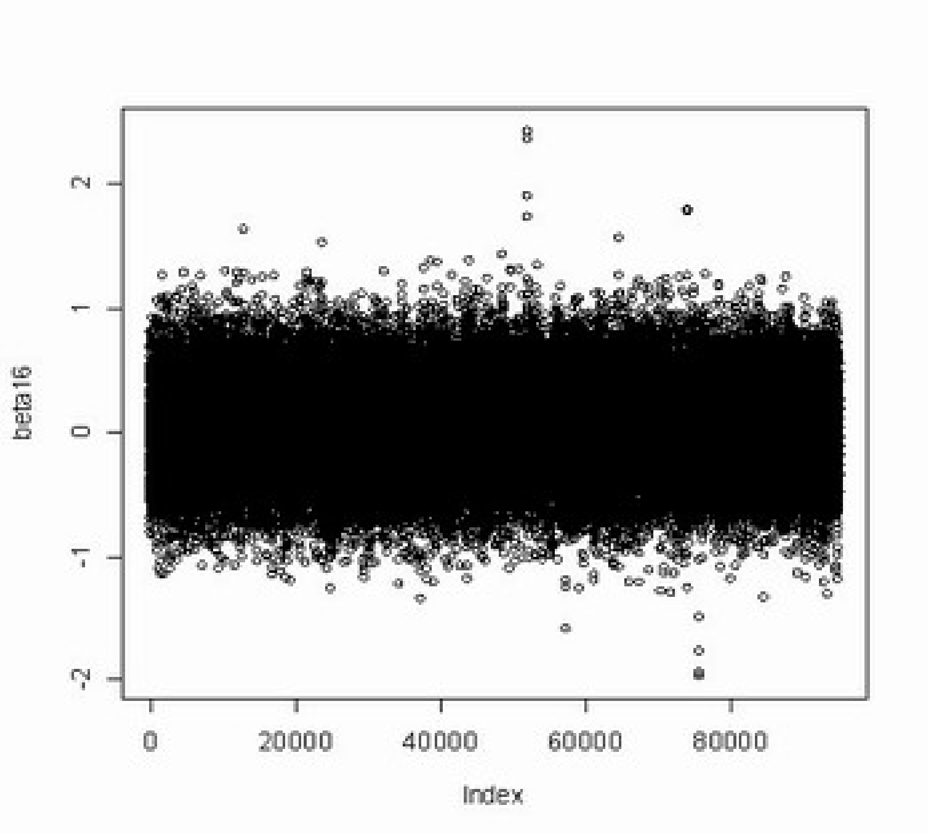

Figure 4: β_{16} from Gibbs Sampler

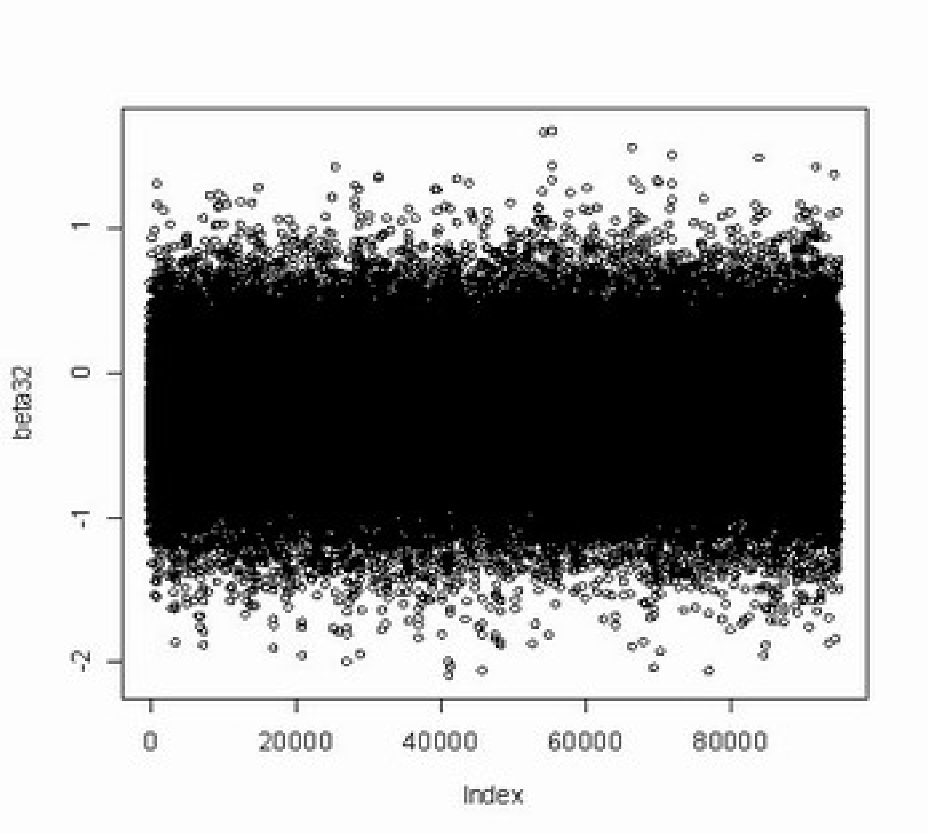

Figure 5: β_{32} from Gibbs Sampler

Detect QTL

Detecting the location on a genome responsible for a quantitative trait is equivalent to selecting the most appropriate model for data. The set of all possible models is denoted by Λ, and we will denote the cardinality or size of Λ by $|\Lambda|$. Thus, we are interested in $p(\delta_K|D)$, the probability of model K given the data. Using Bayes rule, we see that:

$$p(\delta_K|D) = \frac{p(D|\delta_K)p(\delta_K)}{\sum_{K=1}^{|\Lambda|} p(D|\delta_K)p(\delta_K)} \tag{10}$$

Since we assume we have no prior knowledge on which model is the most appropriate, we assign equally likely probabilities to all δ_K's. The quantity $p(D|\delta_K)$ is calculated by

$$p(D|\delta_K) = \int p(D|\delta_K, \lambda_K)p(\lambda_K|\delta_K)d\lambda_K \tag{11}$$

However, this integral is computationally intensive and may be estimated via Monte Carlo methods by

$$\int p(D|\delta_K, \lambda_K)p(\lambda_K|\delta_K)d\lambda_K \approx \frac{1}{t}\sum_{i=1}^{t} p(D|\lambda_K^{(i)}, \delta_K)p(\lambda_K^{(i)}|\delta_K) \tag{12}$$

Where $\lambda_K^{(i)}$, $i = 1, ..., t$ are sample from the posterior distribution. We can use this information to calculate the activation probability defined as $p(\beta_{Kj} \neq 0|D)$ where

$$p(\beta_j \neq 0|D) = \sum_{K=1}^{|\Lambda|} p(\beta_j \neq 0|\delta_K, D)p(\delta_K|D) \tag{13}$$

However, to calculate the activation probability for each β would means that 2^M models need to be created. This may become computationally challenging, so we will define a search algorithm that sequentially divides the genome into smaller and smaller segments until segments with QTLs are identified.

The algorithm first divides the genome into chromosomes. The Bay × Sha population has five chromosomes which we label as a, b, c, d, and e. In this case, we have 2^5 models which need to be fit and their corresponding $p(\delta_K|D)$ calculated with (9), (10), and (11). Using (12) the activation probability of each chromosome can be computed. Table 1 shows the activation probability for each chromosome. The search algorithm then identifies areas or regions of interest as those with activation probabilities greater than 0.5. Since the activation probability of chromosome a and c are more than 0.5, we divide each of the chromosome a and c into two parts and find the activation probability for each segment by the same procedure. As Table 2 shows, the activation probability of segment 1 and segment 2 of chromosome a, $a1$ and $a2$, are more than 0.5. Thus we further divide each of $a1$ and $a2$ into two parts. So we get segments $a11$, $a12$, $a21$ and $a22$. Table 3 shows the activation probability of these four segments. With the same processing, each of $a11$ and $a12$ is divided into two pieces. Table 4 shows the activation probability of each segment. Segment $a122$ represents marker 4 of chromosome a has the only activation probability larger than 0.5, so we conclude that the QTL is on marker 4 of chromosome a.

Table 1: Activation probability of each chromosome

Chromosome	Activation Probability
a	1.0000
b	0.3976
c	0.6026
d	0.3972
e	0.0003

Table 2: Activation probability of segments from first and third chromosomes

Segments	Activation Probability
a1	1.0000
a2	0.9364
c1	0.0634
c2	0.0631

Table 3: Activation probability of segments from first chromosome

Segments	Activation Probability
a11	0.8185
a12	0.9273
a21	0.1147
a22	0.1086

Table 4: Activation probability of first four markers

Segments	Activation Probability
a111	0.0416
a112	0.0143
a121	0.0837
a122	1.0000

CONCLUSION

To utilize existing software, most biologists take the average value of the quantitative trait within each line to perform plant QTL analysis. Therefore, important information about the variability within and between each line is lost. The Bayesian Hierarchical Regression model which can incorporate information of extra level variations of quantitative trait lines is an effective method to detect QTL. We applied this method to a simulated data set from the line information of Bay-0 × Shahdara population in which the QTL was located on the fourth marker of the first chromosome. We used the Bayesian Hierarchical Regression model to model the data set and compare models. The activation probability was calculated to determine which β's are most important for controlling the Quantitative Trait. Since fitting every possible model would be computationally challenging, we constructed a conditional search algorithm that systematically divides segments on the genome into smaller and smaller segments until QTLs are identified. The simulated data set had a QTL located on the fourth marker of the first chromosome and was identified via our Bayesian Hierarchical Regression model.

Although the QTL is detected in the simulated data set, a few issues remain for the further investigation. (1) A sensitivity analysis should be done on the variance of the β's. We need ascertain how our model output depends upon the variance of the β's. This is an important method for checking the quality of our model. (2) Trying different starting points to evaluate our method. (3) Applying this method to the real data set.

REFERENCES

[1] Karl W. Broman “A model selection approach for the identification of quantitative trait loci in experimental crosses”, *Journal of Royal Statistical Society*, 64, Part 4, pp.641-656, 2002.

[2] E. S. Lander, D. Botsterin “Mapping Mendalian factors underlying quantitative traits using RFLP linkage maps”, *Genetics*, 121:185-199, 1989.

[3] Berry CC. , “Computationally Efficient Bayesian QTL Mapping in Experimental Crosses ”, *ASA Proceedings of the Biometrics Section*, pp.164-169, 1998.

[4] Broman KW, Wu H, Sen S,“QTL Mapping in experimental crosses”, *Bioinformatics*,19:889-890, 2003.

[5] Andrew Gelman, John B. Carlin, Hal S. Stern and Donald B. Rubin, *Bayesian Data Analysis*, 2 edn, Chapman Hall/CRC, Boca Raton London NewYork Washington,D.C., 2004.

[6] Karl W. Broman, “The Genomes of Recombinant Inbred Lines”, *Genetics Society of America*, DOI:10.1534/genetics.104.035212, 2004.

[7] Loudet O, Chaillou S, Daniel-Vedele F “Bay-0×Shahdara recombination inbred line population: a powerful tool for the genetic dissection of complex traits in Arabidopsis”, *Theoretical and Applied Genetics* , Vol. 104, 1173-1184, 2002.

[8] Edward L. Boone, Keying Ye, Eric P. Smith “Evaluating the Relationship Between Ecological and Habitat Conditions Using Hierarchical Models”, *Journal of Agriculture, Biological, and Environmental Statistics*, Vol. 10, Number 2 Page 1-17, 2005.

[9] Bernd A. Berg “Markov Chain Monte Carlo Simulations and Their Statistical Analysis ”,*World Scientific* , ISBN 981-238-935-0, 2004.

APPENDIX A

SAS and Fortune Program for QTL

FULL MODEL

SAS code:

```
proc iml;
  use x;
      read all var('x1':'x99') into xx;
  use y;
      read all var('y1':'y10') into yy;
  use data;
      read all var('beta1':'beta99') into beta;
      read all var('theta1':'theta162') into theta;
      read all var('sigma1':'sigma162') into sigma;
      read all var('tau') into tau;
  n=nrow(beta);
  L=nrow(yy);
  ni=ncol(yy);
  start fmodel(tau0,sigma0,n,L,ni,beta,sigma,theta,tau,xx,yy);
       res=j(n,1,0);
         do i=1 to n;
            term1=(tau0+2+L)#log(tau[i,]);
            term2=(ni+sigma0+2)#sum(log(sigma[i,]));
            term3=sum(1/sigma[i,]);
            term4=beta[i,]*beta[i,]`/10000;
                  m=theta[i,]`-xx*beta[i,]`;
            term5=(1+m`*m)/tau[i,];
                  p=j(L,ni,0);
              do j=1 to ni;
                 p[,j]=(yy[,j]-theta[i,]`)##2/sigma[i,]`;
              end;
            term6=sum(p);
            res[i,]=-0.5*(term1+term2+term3+term4+term5+term6);
          end;
        result=sum(exp(res-max(res)))/n;
       return(result);
   finish;
   mm=fmodel(2,2,n,L,ni,beta,sigma,theta,tau,xx,yy);
   print mm;
quit;
```

Fortune code:

```
program Gibbs
    USE MSIMSL
    PARAMETER (M=39,L=165,taunot=0.5,sigmanot=0.5,KK=100000,
    &           kutoff=2000)
 !              M is number of Markers (column) and L is number of lines
    DOUBLE PRECISION  betas(M),XTX(M,M),X(L,M),XB(L),tau2(1),
     &                  taua,taub(1),sigmab(L),Y(L,12),betamu(M),
     &                  covarbeta(M,M),sigma2(L),thetamu(L),thetas(L),
     &                  thetasig(L),ybar(L),sumy(L),RSIG(M,M),TOL,
     &                  stdtau2(1),betasst(M),stdsig(L),ybar2(L),
     &                  stdtheta(L),sigmaa(L),sumy2(L),minloglik,
     &                  liktemp(KK),temp4,temp5,maxloglik,
    &                   sumtemp4,bayesfac,
    &                   yregress(1620),xregress(1620,M),SST,SSE,
    &                   savebeta5(KK), savebeta6(KK), savebeta16(KK),
    &                   savebeta32(KK)
      INTEGER ni(L),IRANK
        !Setting parameters
    taua = taunot + (L/2)
    TOL = 100.0*DMACH(4)
    minloglik = 1.d8
    maxloglik =  -1.d8
    sumtemp4 = 0.d0
    NOBS = 0
      do i = 1,L
        sigmaa(i)=(ni(i)/2) + sigmanot
    enddo
        !Read data
    do i = 1, L
        ni(i) = 10
    enddo
        open(16, file=’bayxsha2.csv’, status=’old’)
        do i=1,L
            read(16,*) (X(i,j),j=1,M)
        enddo
        close(1)
        open(19, file=’newy.csv’, status=’old’)
```

```
        do i=1,L

            read(19,*) (Y(i,j), j=1,ni(i))

        enddo

        close(19)

        do i=1,L

        sumy(i) = 0.d0

        sumy2(i) = 0.d0

        NOBS = NOBS + ni(i)

        end do

        do i=1,L

        do j=1,ni(i)

            sumy(i) =sumy(i) + Y(i,j)            !Create ybar

            sumy2(i) = sumy2(i) + Y(i,j)*Y(i,j)

        enddo

        ybar(i) = sumy(i)/ni(i)

        thetas(i) = ybar(i)

        sigma2(i) = (sumy2(i) - ni(i)*(ybar(i)**2))/(ni(i) - 1)

        if (sigma2(i).eq.0.d0) sigma2(i) = 1.d0

        ybar2(i) = sumy2(i)/ni(i)

        enddo

        do i = 1,L

            sumtheta = sumtheta + thetas(i)

            sumtheta2 = sumtheta2 + (thetas(i)**2)

        enddo

        thetabar = sumtheta/L

        tau2 = (sumtheta2 - L*(thetabar**2))/(L - 1)

c       Getting ready for regression
```

```
    num = 1
    do i = 1,L
        do j = 1,ni(i)
            yregress(num) = Y(i,j)
            num = num + 1
        enddo
    enddo
    num2 = 1
    do i = 1,L
        do k = 1,ni(i)
            do j = 1,M
                xregress(num2,j) = X(i,j)
            enddo
            num2 = num2 + 1
        enddo
    enddo
CALL DRLSE (NOBS, yregress, M, xregress, NOBS, 0, betas,
&       SST, SSE)
    CALL DMXTXF (L, M, X, L, M, XTX, M)          !Calculates XTX
    CALL DMURRV (L, M, X, L, M, betas, 1, L, XB) !Mult matrix x vector
    !Gibbs Sampler
    do k=1,KK
!*****  THETAS      ***************************
    CALL thetapar (tau2,sigma2,XB,L,ybar,ni,thetamu,thetasig) !parameter
    CALL DRNNOR (L,stdtheta)
    do i=1,L
    thetas(i) = stdtheta(i)*thetasig(i) + thetamu(i)
```

```
        enddo
    !*****  TAU         ***************************
        CALL tauparm (thetas,XB,L,taub)
        CALL drngam(1,taua,stdtau2)
        tau2(1) = taub(1)/stdtau2(1)
    !*****  BETA        ***************************
        CALL betapar (XTX,M,tau2,L,thetas,X,betamu,covarbeta)
        CALL DCHFAC (M, covarbeta, M, TOL, IRANK, RSIG, M)   ! Cholesky factor
        CALL DRNMVN (1, M, RSIG, M, betasst, M)
        do i=1,M
        betas(i) = betasst(i) + betamu(i)
        enddo
        CALL DMURRV (L, M, X, L, M, betas, 1, L, XB) !Mult matrix x vector
!   *****   SIGMA       ***************************
        CALL sigmaparm (ybar,ybar2,ni,thetas,L,sigmab)
        CALL drngam(L,sigmaa(1),stdsig)
        do i = 1,L
        sigma2(i) = sigmab(i)/stdsig(i)
        enddo
        savebeta5(k) = betas(5)
        savebeta6(k)=betas(6)
        savebeta16(k) =betas(16)
        savebeta32(k)=betas(32)
     CALL llike (betas,XB,tau2,Y,sigma2,thetas,
     &      L,M,sigmaa,taua,temp4,temp5)
     liktemp(k)=temp4
     if ((temp5.ge.maxloglik) .and. (k.ge.kutoff)) maxloglik = temp5
     if ((temp5.le.minloglik) .and. (k.ge.kutoff)) minloglik = temp5
    enddo
         open(100,file='savebeta.csv',status='new')
         do nnum = 1,KK
         write(100,*) savebeta5(nnum),savebeta6(nnum),
    &    savebeta16(nnum),savebeta32(nnum)
         enddo
         close(100)
    do k=(kutoff+1),KK
        sumtemp4 = sumtemp4 + liktemp(k)
    enddo
    bayesfac = sumtemp4/(KK-(kutoff+1))
    open(50,file='Bayesoutput.txt',status='new')
    write(50,*) "Sumtemp 4 = ",sumtemp4
    write(50,*) "Bayes factor = ", bayesfac
    write(50,*) "Minimum log-likelihood = ", minloglik
    write(50,*) "Maximum log-likelihood = ", maxloglik
    close(50)
    end
!   SUBROUTINES
    SUBROUTINE tauparm (thetas,XB,L,taub)
    DOUBLE PRECISION sumTXB,taub(1),thetas(L),XB(L)
    INTEGER L
           sumTXB=0.d0
           do i=1,L
            sumTXB=sumTXB + (thetas(i) - XB(i))*(thetas(i) - XB(i)) +1
         enddo
        taub(1)=0.5*sumTXB
           end
    SUBROUTINE sigmaparm (ybar,ybar2,ni,thetas,L,sigmab)
    DOUBLEPRECISION ybar(L),thetas(L),sumythetas,sigmab(L),ybar2(L)
    INTEGER ni(L)
        sumythetas=0.d0
        do i=1,L
        sigmab(i) = 0.5*(1+(ni(i)*ybar2(i) - 2*thetas(i)*ni(i)*ybar(i)
    &      + ni(i)*thetas(i)*thetas(i)))
        enddo
        end
    SUBROUTINE betapar (XTX,M,tau2,L,thetas,X,betamu,covarbeta)
    DOUBLE PRECISION    XTX(M,M),step1(M,M),covarbeta(M,M),mupart2(M),
    &   thetas(L),betamu(M),tau2(1),X(L,M)
    INTEGER M,L
        do i=1,M
        do j=1,M
        if (i.eq.j) then
        step1(i,j)=(1/100)+((1/tau2(1))*XTX(i,j))
            else
                step1(i,j) =  ((1/tau2(1))*XTX(i,j))
        endif
        enddo
        enddo
        CALL DLINDS (M, step1, M, covarbeta, M)
        CALL DMURRV (L, M, X, L, L, thetas, 2, M, mupart2)
        do i = 1,M
```

```
        mupart2(i) = mupart2(i)/tau2(1)
    enddo
    CALL DMURRV (M, M, covarbeta, M, M, mupart2, 1, M, betamu)
    end
    SUBROUTINE thetapar (tau2,sigma2,XB,L,ybar,ni,thetamu,thetasig)
    DOUBLE PRECISION tau2(1),sigma2(L),XB(L),ybar(L),thetamu(L),
&       thetasig(L)
    INTEGER L ,ni(L)
    do i=1,L
    thetamu(i) = (1/tau2(1))*(tau2(1)*sigma2(i)/(ni(i)*tau2(1)
&   +sigma2(i)))*XB(i) +(1/sigma2(i))
&   *(tau2(1)*sigma2(i)/(ni(i)*tau2(1)+sigma2(i)))*
&   ni(i)*ybar(i)
    enddo
    do i=1,L
        thetasig(i) = sqrt(tau2(1)*sigma2(i)/(ni(i)*tau2(1)
&   +sigma2(i)))
    enddo
    end
   SUBROUTINE llike (betas,XB,tau2,Y,sigma2,thetas,
 &      L,M,sigmaa,taua,flik,likehood2)
       DOUBLE PRECISION  betas(M),XB(L),tau2(1),
 &                  taua,Y(L,10),btb,thetas(L),
 &                  sigma2(L),sigmaa(L),lik1,lik2,likehood,flik,
&                   likehood2
   INTEGER M,L
   lik1=0.d0
   lik2=0.d0
   btb=0.d0
do i=1,L
    lik1= lik1 - (sigmaa(i))*dlog(sigma2(i)) -
 &  (1/(2*sigma2(i))) -
 &  (1/(2*tau2(1)))*
 &  (thetas(i) - XB(i))*
 &  (thetas(i) - XB(i))
     end do
    do i=1,L
    do j=1,10
        lik2 = lik2 -(1/(2*sigma2(i)))*(Y(i,j)-thetas(i))*
&       (Y(i,j)-thetas(i))
    end do
    end do
    do i = 1,M
    btb=btb + betas(i)*betas(i)
    end do
    likehood = lik1 + lik2 - (taua)*dlog(tau2(1))
 &  - (1/(2*tau2(1))) - (1/200) * btb
    likehood2=likehood +500  !Adjusting likelihood
    flik = dexp(likehood2)
end
```