STATISTICAL METHODS FOR GENETIC RISK CONFIDENCE INTERVALS, BAYESIAN DISEASE RISK PREDICTION, AND ESTIMATING MUTATION SCREENING SATURATION

by

Ying Shan

BS of Biomedical Engineering,

South China University of Technology, China, 2012

Submitted to the Graduate Faculty of

the Graduate School of Public Health in partial fulfillment

of the requirements for the degree of

Doctor of Philosophy

University of Pittsburgh

2016

UNIVERSITY OF PITTSBURGH

GRADUATE SCHOOL OF PUBLIC HEALTH

This dissertation was presented

by

Ying Shan

It was defended on

7/12/2016

and approved by

Daniel E. Weeks, PhD

Professor

Department of Human Genetics

Graduate School of Public Health

University of Pittsburgh

Eleanor Feingold, PhD

Professor

Department of Human Genetics

Graduate School of Public Health

University of Pittsburgh

Roger S. Day, ScD

Associate Professor

Department of Biomedical Informatics

School of Medicine

University of Pittsburgh

Yong Seok Park, PhD

Assistant Professor

Department of Biostatistics

Graduate School of Public Health

University of Pittsburgh

Wei Chen, PhD

Assistant Professor

Department of Pediatrics

School of Medicine

University of Pittsburgh

Dissertation Director: **Daniel E. Weeks**, PhD

Professor

Department of Human Genetics

Graduate School of Public Health

University of Pittsburgh

STATISTICAL METHODS FOR GENETIC RISK CONFIDENCE INTERVALS, BAYESIAN DISEASE RISK PREDICTION, AND ESTIMATING MUTATION SCREENING SATURATION

Ying Shan, PhD

University of Pittsburgh, 2016

ABSTRACT

Genetic information can be used to improve disease risk estimation as well as to estimate the number of genes influencing a trait. Here we explore these issues in three parts. 1) For an informed understanding of a disease risk prediction, the confidence interval of the risk estimate should be taken into account. But few previous studies have considered it. We propose a better risk prediction model and provide a better screening strategy considering the confidence intervals. Risk models are built with varying numbers of genetic risk variants known as single nucleotide polymorphisms (SNPs). Inclusion in the risk model of SNPs, sorted in decreasing order by effect size, with smaller effects modestly, shifts the risk but also increases the confidence intervals. The more appropriate risk prediction model should not include the small effect SNPs. The newly proposed screening method is superior to the traditional one as evaluated by net benefit quantity. 2) Many methods have been developed for associated SNP selection, SNP effect estimation, and risk prediction. A Bayesian method designed for continuous phenotypes, BayesR, shows good characteristics. Here, we developed an extension of BayesR (BayesRB), so that the method can be used for binary phenotypes. For SNP effect estimation, BayesRB shows the unbiasedness and sparseness for the big and small effect SNPs, respectively. It also performs well on risk prediction, but not on associated SNP selection. 3) When a recessive forward genetic screening study (RFGSS) is carried out to detect disease mutations, it is important to estimate the screening saturation so as to

guide the screening strategy. Here, we develop a simulation-based "unseen species" method to estimate the screening saturation in a RFGSS. We simulated a RFGSS process based on a real study and compared our method to both nonparametric methods and parametric methods. The proposed method performs better than all the other methods, except an existing "unseen species" method. The above three newly proposed methods are helpful for constructing better risk prediction models and for estimating the number of disease contributing genes. These methods can be applied to different disease studies and may make contributions to public health.

Keywords: Risk prediction, Confidence intervals, Bayesian models, Screening saturation.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

1.0 SPECIFIC AIMS

This research aims to apply genetic statistical methods to solve three research questions in genetics and shed more light on the statistical methodologies applicable to these questions. The approaches aim to achieve the following goals: 1) construct a better risk prediction model and provide a better screening strategy by taking the confidence interval of the estimated risk into account using the reported single nucleotide polymorphisms (SNPs); 2) develop a Bayesian based approach to detect the associated SNPs, estimate the SNP effects and predict the risks of being affected by a dichotomous phenotype taking into account the effects of all the SNPs; 3) develop a simulation based "unseen species" method to estimate the total number of disease genes in a recessive forward genetic screening study (RFGSS).

Aim 1. Genetic risk models and screening strategies: Influence of model size on risk estimates and precision

To better construct the risk prediction model and to decide whom to screen, the confidence interval of the disease risk estimate should be taken into account. But few previous studies have done it. In aim 1, we constructed a better risk prediction model and provided a better screening strategy by taking the confidence interval of the predicted risk into account. We studied how risks shift and confidence intervals change as the model sizes (e.g., number of SNPs in the risk model) increase when adding more SNPs in the order of decreasing effect size. This was done using simulated data set and two real data sets of abdominal aortic aneurysms (AAA) and age-related macular degeneration (AMD). We found that if we order SNPs by their effect sizes and build risk models of various sizes by adding in the next weakest SNP, risk shifts between successive models becomes more and more modest, and the

confidence intervals of the risk estimates tend to become larger. The best risk prediction model may not be the model with the biggest model size. We also provided a new screening strategy taking account the confidence intervals of the risk estimates. We compared it with the traditional screening strategy using net benefit quantity. The new screening strategy performs better than the traditional one with lower false negative rates.

Aim 2. Develop a Bayesian approach (BayesRB) to estimate the SNP effects and predict genetic risk for dichotomous traits

Many methods have been developed for associated SNP selection, SNP effect estimation, and risk prediction. One of the most popular methods is the genome-wide association study (GWAS). In GWAS, for dichotomous traits, each SNP effect is typically estimated from a marginal logistic regression model, which usually only contains the SNP and the covariates of the clinical traits. It consequently fails to account for the effects of other SNPs, which decreases the precision and the power to detect true associations [Moser et al., 2015]. BayesR [Moser et al., 2015] is a more accurate method, which detects the associated SNPs, estimates the SNP effects and makes the prediction of phenotypes based on all SNPs simultaneously. It shows good characteristics of unbiasedness, accuracy, sparseness and robustness. However, this method was designed to be applied to the continuous phenotype data sets. In order to detect the associated SNPs, estimate the SNP effects and make the prediction of the risk for the dichotomous phenotype with all the good characteristics that BayesR has, we made an extension of the BayesR method, called BayesRB, by adding auxiliary variables to the BayesR model. We applied the BayesRB to pilot simulated data sets to assess convergence and make sure that BayesRB works. Then, we simulated 50 genome-wide simulated data sets. We applied BayesRB to the genome-wide simulated data sets to evaluate BayesRB's performance of associated SNP identification, SNP effect estimation and the risk prediction and compare its performance to BayesR, LASSO and logistic regression. Since BayesR was applied to the Welcome Trust Case Control Consortium (WTCCC) data set in Moser et al., we also applied BayesRB to it and compared their performance. For SNP effect estimation, BayesRB has similar estimates to the logistic regression for big effect SNPs, and shows

BayesR's sparseness characteristic for small effect SNPs. It also has better risk prediction performance than logistic regression and LASSO, but worse than BayesR. For associated SNP selection, BayesRB does not perform better than the other three methods.

Aim 3. Develop a simulation-based "unseen species" method to estimate the total number of disease genes in a recessive forward genetic screening study (RFGSS).

Recessive forward genetic screening study (RFGSS) is widely conducted to discover the disease mutations and detect the disease etiology. Estimating the screening saturation in a RFGSS guides the screening strategy. To our knowledge, no previous study has estimated the screening saturation and the total number of disease genes in a RFGSS before. In aim 3, we developed a simulation-based "unseen species" method to estimate the total number of disease genes in the RFGSS.

We simulated a RFGSS process based on a real study and applied the proposed method to the simulated data sets to estimate the total numbers of disease genes and their quantile intervals. We evaluated the unbiasedness and precision of the estimates. We evaluated the quantile intervals by the coverage rates. In the RFGSS scenario, we compared the performance of the proposed method to both nonparametric methods and parametric methods. The proposed method performs better than all the other methods except an existing "unseen species" method, with mean closest to the truth and with relatively small standard deviation.

2.0 AIM 1. GENETIC RISK MODELS AND SCREENING STRATEGIES: INFLUENCE OF MODEL SIZE ON RISK ESTIMATES AND PRECISION

The majority of the text in this chapter comes from a manuscript [Shan et al., 2016] submitted to Genetic Epidemiology.

2.1 BACKGROUND

Personalized genomics is currently a widely discussed topic [Bloss et al., 2011]. Personalized genomics companies and many publications [Evans et al., 2009; Morrison et al., 2007; Wray et al., 2007] have provided disease risk prediction models based on genetic predictors. However, these risk reports seldom take the confidence interval of the risk estimate into account [Kalf et al., 2014]. For example, 23andMe presented to its customers a point estimate of the risk and the average risk of the disease in the population, as well as how much higher the estimated risk was than the average risk. 23andMe did not present confidence intervals of the provided risk estimates. In many publications, especially when risk estimates are based on odds ratios derived by meta-analysis, the confidence intervals of the risk estimates are not presented nor considered in the evaluation of the risk model. Many studies have applied regression models to a set of risk single-nucleotide polymorphisms (SNPs) to make predictions. Using the area under the curve (AUC) metric to evaluate their risk models, they conclude that the more risk SNPs in the risk model, the larger the AUC will be, thus, the better the ability to predict the risk [De Jager et al., 2009; van Dieren et al., 2012]. However, as we illustrate here, as the number of risk SNPs in the model increases, the confidence interval of the risk estimate can widen. In fact, a risk estimate with a larger confidence interval from a

larger model with more SNPs may not be practically better than a similar risk estimate with a smaller confidence interval from a smaller model based on fewer SNPs. When presenting and evaluating risk estimates, it is important to consider the level of uncertainty in the risk estimate.

In this study, we explore the changes of risk estimates and their 95% confidence interval widths as more SNPs, in the order of decreasing effect size, are added into the model, based on both simulated and real data. We also created a reclassification table to evaluate the effect of the added SNPs predictors, taking the confidence interval of the risk estimate into account. Finally, we selected the best screening strategy based on the net benefit quantity and the reclassification rate.

2.2 METHODS

2.2.1 Data Description

In this study, we use three data sets to evaluate and compare our risk models. The first data set is a simulated one. We simulated a data set of 100,000 people assuming a genetic model based on 19 independent risk SNPs with odds ratios and allele frequencies matching those observed in a large meta-analysis of age-related macular degeneration (AMD) [Fritsche et al., 2013], using the Multiple Gene Risk Prediction Performance (mgrp) R package [Pepe et al., 2010a]. The reason for the independency of the 19 SNPs is that in most real studies, the top risk SNPs are far enough from each other to be considered independent. In the large meta-analysis of AMD, the 19 SNPs were shown to be highly related to AMD. AMD is a progressive neurodegenerative disease, which constitutes one of the primary causes of visual impairment and irreversible blindness in the elderly of western countries [Klein et al., 2011]. In our simulation, we assumed that the disease is dichotomous with a prevalence of 0.055, which is similar to the prevalence of AMD.

The second data set is from a study of abdominal aortic aneurysms (AAA). AAA is the most common form of aortic aneurysm. In general, the prevalence of AAA 2.9 to 4.9 cm

in diameter ranges from 1.3% for men aged 45 to 54 years to up to 12.5% for men 75 to 84 years of age. Comparable prevalence figures for women are 0% and 5.2%, respectively [Rooke et al., 2011]. Up to 10% of the male population who are more than 65 years old has AAA, and 80-90% of ruptures lead to sudden death [Assar and Zarins, 2009]. Our goal was to classify the population into high-risk and low-risk categories, where "high risk" is defined as having a risk higher than the population prevalence. Our motivation was to identify people with high AAA risk for targeted ultrasound screening. The samples were genotyped at 731K SNPs using the Illumina OmniExpress platform (dbGaP Study Accession numbers: phs000381.v1.p1, phs000408.v1.p1 and phs000387.v1.p1). AAA cases and controls were identified by electronic phenotyping [Borthwick et al., 2015]. After imputation and quality control [Verma et al., 2014], 2,626 samples (733 cases and 1,893 controls) were available. The imputed data are part of the eMERGE Network Imputed GWAS data for 41 Phenotypes (the dbGaP eMERGE Phase 1 and 2 Merged data Submission) with an accession number phs000888.v1.p1. By modeling in a much larger electronic medical record (EMR)-based clinical data set, seven easy-to-measure clinical predictors (age, smoking status, sex, systolic blood pressure, diastolic blood pressure, height and weight) were chosen for use in our risk models [Smelser et al., 2014]. Based on prior literature [Biros et al., 2011; Bown et al., 2011; Elmore et al., 2009; Galora et al., 2013; Giusti et al., 2008; Harrison et al., 2013; Helgadottir et al., 2012; Jones et al., 2013; Saracini et al., 2012],15 SNPs present in the imputed data were selected with odds ratios in the literature ranging from 0.41 to 2.16 (Table 2.1).

The third data set is from a study of the genetics of AMD [Weeks et al., 2004, 2000]. In our analysis, for 1,015 unrelated individuals (882 cases and 133 controls) high quality genotypes were available at 14 of the 19 SNPs mentioned above, and these 14 were used as predictors in the AMD data analysis. The cases in our study were defined according to the diagnosis criteria of "Model C" as defined in Weeks, et al. [2004]. Under Model C, cases are those who are definitely or probably affected with AMD or with a related maculopathy. Model C also included individuals with endstage disease, in the absence of any other documentation of macular pathology. The controls had no AMD symptoms with an age at last exam ≥ 65.

Table 2.1: **The original odds ratios (OR), the odds ratios used for ordering, and the allele frequencies of the predictors for Abdominal Aortic Aneurysm (AAA) data analysis.**

a. Odds ratios and the adjusted odds ratios of the genetic predictors.

Genetic Predictors	Original OR in prior study	Adjusted OR	Risk allele frequencies
rs8003379	0.41	2.44	0.238
rs3781590	2.16	2.16	0.335
rs4988300	2.16	2.16	0.486
rs326118	0.47	2.13	0.113
rs764522	1.69	1.69	0.192
rs5186	1.60	1.60	0.300
rs679620	1.48	1.48	0.482
rs2252070	1.37	1.37	0.305
rs4353	1.35	1.35	0.462
rs7635818	1.33	1.33	0.458
rs599839	0.81	1.23	0.223
rs3798220	1.23	1.23	0.019
rs7529229	0.84	1.19	0.393
rs12039875	1.18	1.18	0.323
rs1466535	1.15	1.15	0.321

b. Odds ratios and the adjusted odds ratios of the clinical predictors.

Clinical Predictor	OR from EMR	Adjusted OR
Age (years)a	1.51	1.51
Ever smoker	3.87	3.87
Male sex	1.91	1.91
Systolic blood pressure (mmHg)	1.40	1.40
Diastolic blood pressure (mmHg)	1.56	1.56
Height (ft)	3.83	3.83
Weight (lb)	1.09	1.09

The odds ratios used for ordering the SNPs are obtained by inverting any original odds ratios that are below 1 to be above 1. The clinical predictors were chosen based on Smelser et al. [2014].

Table 2.2: **The odds ratios and the risk allele frequencies of the SNPs in the simulation study.**

SNP	Odds Ratio	Risk Allele Frequency
1	2.76	0.297
2	2.43	0.636
3	1.74	0.857
4	1.42	0.202
5	1.31	0.743
6	1.30	0.830
7	1.23	0.097
8	1.22	0.764
9	1.16	0.787
10	1.15	0.512
11	1.15	0.485
12	1.15	0.212
13	1.14	0.312
14	1.13	0.478
15	1.13	0.728
16	1.11	0.614
17	1.10	0.637
18	1.10	0.458
19	1.10	0.443

2.2.2 Data analysis

First, for all the three data sets, we used logistic regression to fit the risk models. To avoid over fitting, we used four fifths of the data as the training data set and the rest of the data as the testing data set. We did not include any covariates besides the SNPs when analyzing the AMD and simulated data sets. But we included seven easy-to-measure clinical predictors (age, smoking status, sex, systolic blood pressure, diastolic blood pressure, height and weight) when analyzing the AAA data set. Using the training data set, we fit the largest model using logistic regression with all the SNPs to estimate an odds ratio for each SNP.

After flipping odds ratios < 1 to be > 1, we ordered the SNPs according to the adjusted odds ratios. We fit successively larger models using the training data set, increasing the model size K by adding in the risk SNP with the next biggest adjusted odds ratio (Table 2.1, Table 2.2). In the training process, we estimated the natural logarithm of odds ratios: β parameters for each model. Then we estimated risks for each individual in the testing data set by plugging in the β's as estimated from the training set using K SNPs. When estimating risks from a case/control sample using logistic regression, the resulting risk estimate is not the absolute risk, but rather depends on the case/control ratio in the sample itself. Accordingly, for the case/control data sets, risk estimates were adjusted using the methods described in Pyke et al. [1979]. For each person in the testing data set, we recorded the risk estimate, its 95% confidence interval, the model size, and the SNP genotypes.

We then explored how the risk shifts as the model size increases using bean plots and risk trajectory plots. To quantify the magnitudes of the risk shifts, we recorded the maximum of the absolute risk shifts (MRS) between model k and all bigger models for each individual. We recorded the maximum, across all individuals, of the MRS when additional SNPs were added to the model k which we refer to as the "maxMRS"; and the 95th percentile of the MRS which we refer to as the "95PMRS". To investigate the relationship of the confidence interval width and the model size, we used Spearman's rank correlation test and bean plots.

For the AAA and AMD data sets, we evaluated the risk models using reclassification tables, taking the confidence interval into account, classifying individuals into high-risk and low-risk groups based on a threshold T corresponding to the population prevalence (we assumed the prevalence was 0.033 for AAA and 0.055 for AMD). In the traditional reclassification tables (which do not take the confidence intervals into account), assignment to either the low or the high risk classes is defined solely based on the risk threshold T. In order to take the risk confidence intervals into account, we created confidence interval augmented (CI-augmented) reclassification tables where we redefined the low*/high* risk classes to contain individuals whose risk estimates were lower/higher than T and whose confidence interval did not overlap T (Figure 2.1a). Individuals in these two classes had risk estimates that were unambiguously either below or above T. Then we added two more classes, denoted as "{-T}" and "{+T}" which contain individuals with risk estimates with confidence intervals

that overlap the threshold T (Figure 2.1a). The individuals in the "{-T}" class had risk estimates $<$ T, while those in the "{+T}" class had risk estimates $\geq$ T. For individuals in these two classes, it is not clear if their true risk is above or below T. As the CI-augmented confidence interval classifies the samples into four categories (low*, {-T}, {+T}, high*), there are three possible screening strategies: 1. screen the samples in high* risk class only (defined as {T,1]); 2. screen the samples in both {+T} class and high* risk class (defined as {+T,1]); 3. screen the samples in {-T}, {+T} and high* risk class (defined as {-T,1]). We calculated the net benefit [McGeechan et al., 2014], which provides a measure of the number of people correctly screened as having the outcome, adjusted for the number of people incorrectly screened as having the outcome. The net benefit formula is:

$$Net\ benefit = \frac{True\ positive}{n} - \frac{False\ positive}{n}(\frac{T}{1-T}),$$

where n is the sample size, and T is the threshold as indicated above. We defined the reclassification rate of "lower risk group⇔higher risk group" as the proportion of individuals reclassified from the lower risk group to the higher risk group or from the higher risk group to the lower risk group. Then we calculated the reclassification rate of [0,-T}⇔{+T,1] and the reclassification rate of low*⇔{-T,1] according to the screening strategies 2 and 3. We also evaluated the rate of correct reclassifications for the three screening strategies. Correct reclassification means reclassifying cases from the lower risk group to the higher risk group, or reclassifying controls from the higher risk group to the lower risk group. We used the net benefit and the rates of correct reclassification to select the best screening strategy. Furthermore, in order to explore the influence of model size on the confidence interval width, we recorded how many confidence interval widths increased and decreased when additional SNPs were added to the initial model.

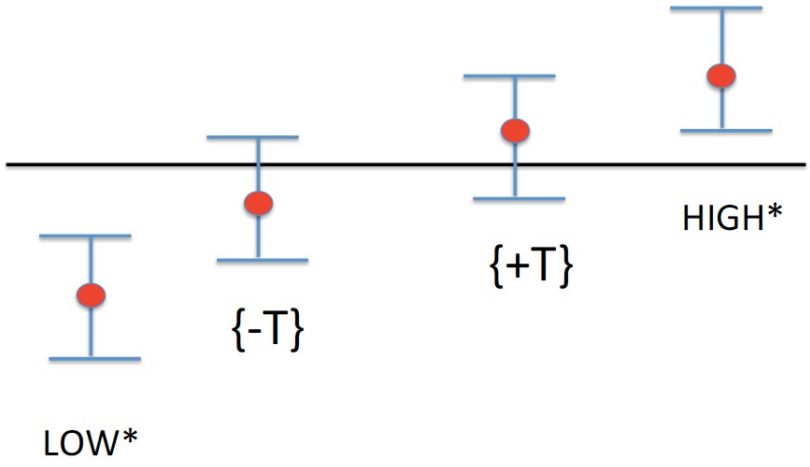


Figure 2.1: **Explanation of the notations of "low*", "{-T}", "{+T}" and "high*" categories.**

2.3 RESULTS

First, in each data set, we examined how much the risk shifted when one more SNP with the next largest adjusted odds ratio was added to the model. To explore the risk shift at the individual level, we plotted representative risk trajectories as SNPs were added to the model in the order of decreasing effect sizes (Figure 2.2). As expected, the risks shift less when SNPs with the smaller odds ratios are added. Figure 2.2 shows, at the individual level, movement in risk among the smaller models has a marked flattening of the risk trajectories as the models get larger. We also found that individuals with higher initial risks tend to have their risks shift more than those with lower initial risks as the model size increases. In the simulated data set (Figure 2.2a), when the three initial risks are 0.027, 0.068 and 0.161, the maxMRS's based on model 1, which is the smallest model with only one SNP, are 0.15, 0.27 and 0.39, and the 95PMRS's based on model 1 are 0.05, 0.11 and 0.21, respectively. In AAA and AMD data set, the 95PMRS's based on model 1 are also bigger when the initial risks are bigger (Figure 2.2b,c).

We then explored the risk shift at the population level, as more SNPs are added into the risk model. Table 2.3 shows that the risks do not shift markedly once the model size is bigger than a certain level. For example, if we let the "maxMRS-selected model" be the smallest model with a maxMRS < 0.06, then in all the three data sets, the 95PMRS of the models bigger than the maxMRS-selected model were all smaller than 0.025. Furthermore, if we let M_i represent the model with i SNPs, in all the three data sets, when the model size is bigger than the maxMRS-selected model, 100% of the M_{i+1} risk estimates lay inside the corresponding M_i confidence interval (Figure 2.3a-c) and 100% of the M_{i+1} confidence intervals overlap with the corresponding M_i confidence interval (Figure 2.3d-e). In addition, when the model size is bigger than the maxMRS-selected model, all the M_{i+1} confidence intervals overlapped more than 50%, 90% and 95% with the corresponding M_i confidence intervals, in the simulation data set, AAA data set and AMD data set, respectively. Consistent with these observations, Figure 2.4 shows that when the model size was greater than the maxMRS-selected model, the risk shift distributions did not change markedly as the model sizes grew.

Table 2.3: **The maxMRS and 95PMRS measures in the simulation data set, AAA data set and AMD data set.**

	Simulation data set		AAA data set		AMD data set	
# of SNPs in Model	maxMRS	95PMRS	maxMRS	95PMRS	maxMRS	95PMRS
1	0.395	0.113	0.221	0.075	0.381	0.227
2	0.311	0.076	0.222	0.070	0.322	0.158
3	0.286	0.067	0.182	0.063	0.307	0.136
4	0.214	0.060	0.133	0.051	0.297	0.128
5	0.209	0.053	0.147	0.041	0.222	0.104
6	0.185	0.050	0.098	0.038	0.164	0.079
7	0.176	0.046	0.070	0.027	0.142	0.066
8	0.163	0.041	**0.058**	0.022	0.096	0.050
9	0.150	0.036	0.042	0.016	0.064	0.035
10	0.119	0.032	0.039	0.014	**0.032**	0.021
11	0.103	0.029	0.030	0.011	0.022	0.010
12	0.099	0.025	0.021	0.009	0.007	0.004
13	0.087	0.022	0.020	0.006	0.002	0.001
14	0.069	0.019	0.017	0.005		
15	0.060	0.016	0.002	0.001		
16	**0.043**	0.012				
17	0.025	0.007				
18	0.011	0.003				

The bold values indicate the "maxMRS-selected model" which is the smallest model with maxMRS less than 0.06.

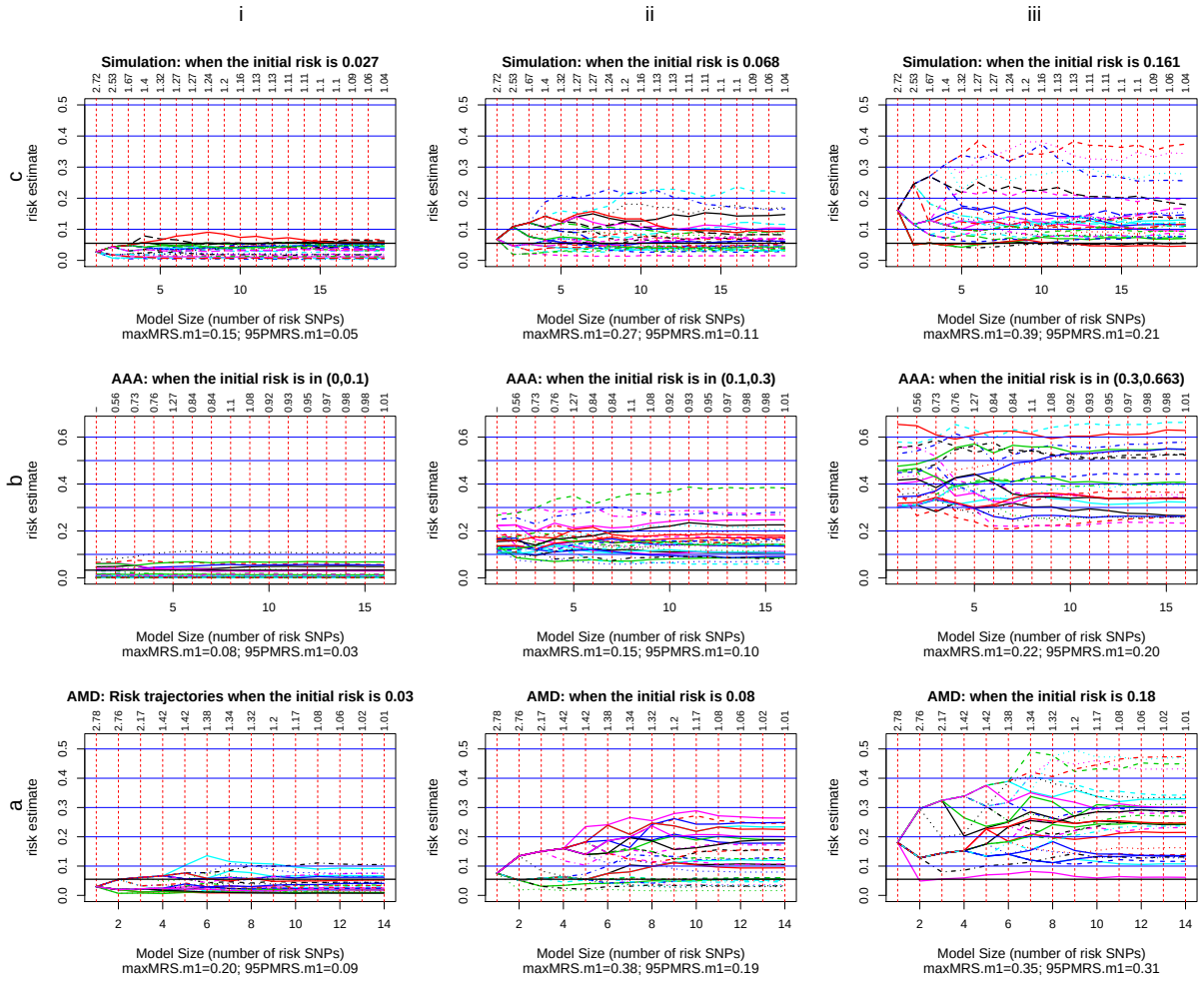


Figure 2.2: **The representative risk trajectories as SNPs are added to the model in the order of decreasing effect sizes.**

Risk trajectories as categorized by the initial risk for (a) the simulation study, (b) the AAA data set, and (c) the AMD data set. Each graph contains the trajectories of 30 individuals randomly chosen from the testing data set. The odds ratios of the added SNPs are shown on the top of each sub figure. The horizontal black line is the disease prevalence (0.055 for the simulated data set and the AMD data set; 0.033 for the AAA data set). The maxMRS.m1 is the maxMRS based on model 1, while the 95PMRS.m1 is the 95PMRS based on model 1.

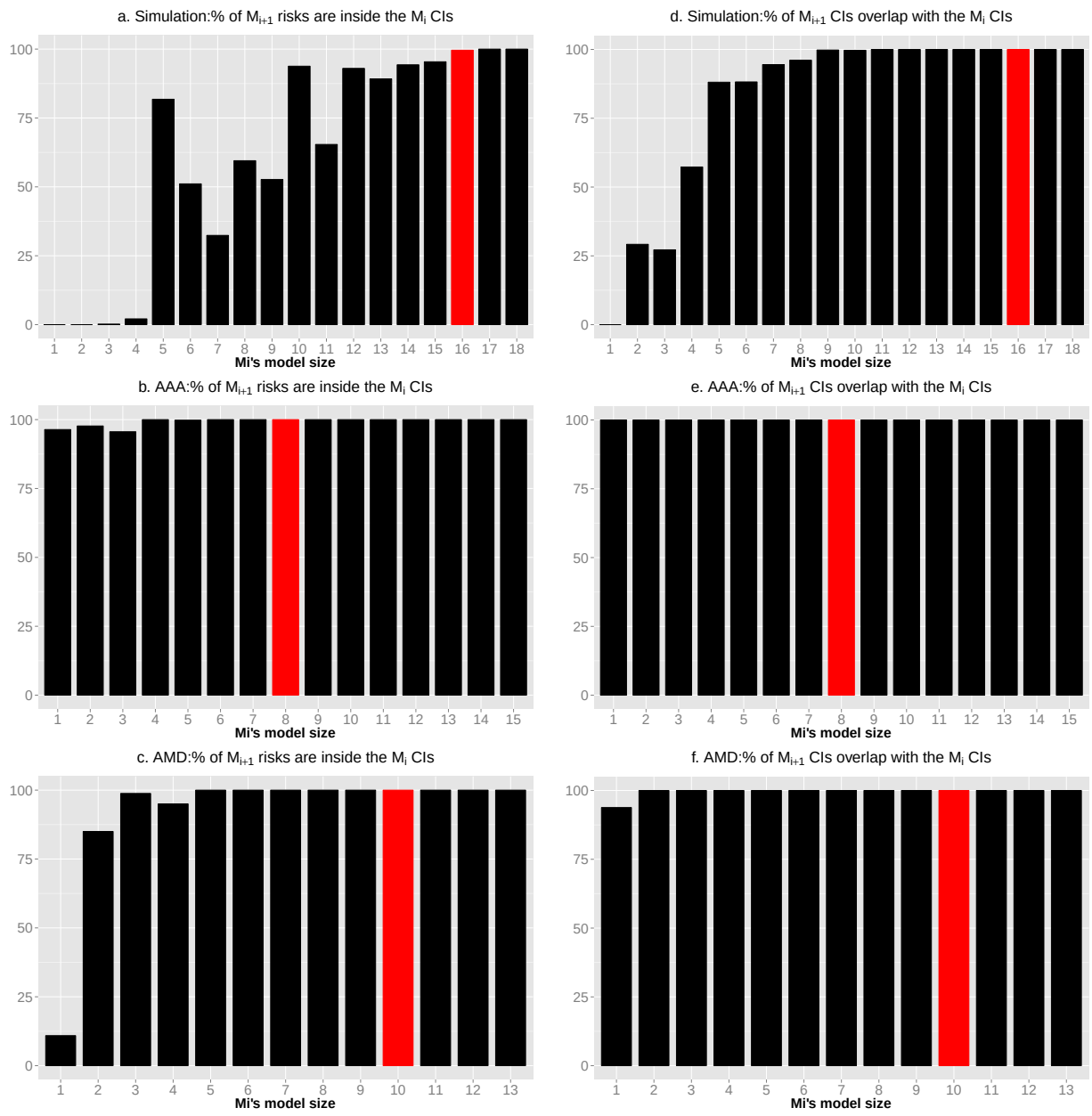


Figure 2.3: **Percentages of M_{i+1} risks inside the M_i confidence intervals (a-c), and percentages of M_{i+1} confidence intervals that overlap with the M_i confidence intervals (d-f), where model M_{i+1} is one SNP larger than model M_i. The red bar shows the maxMRS-selected models.**

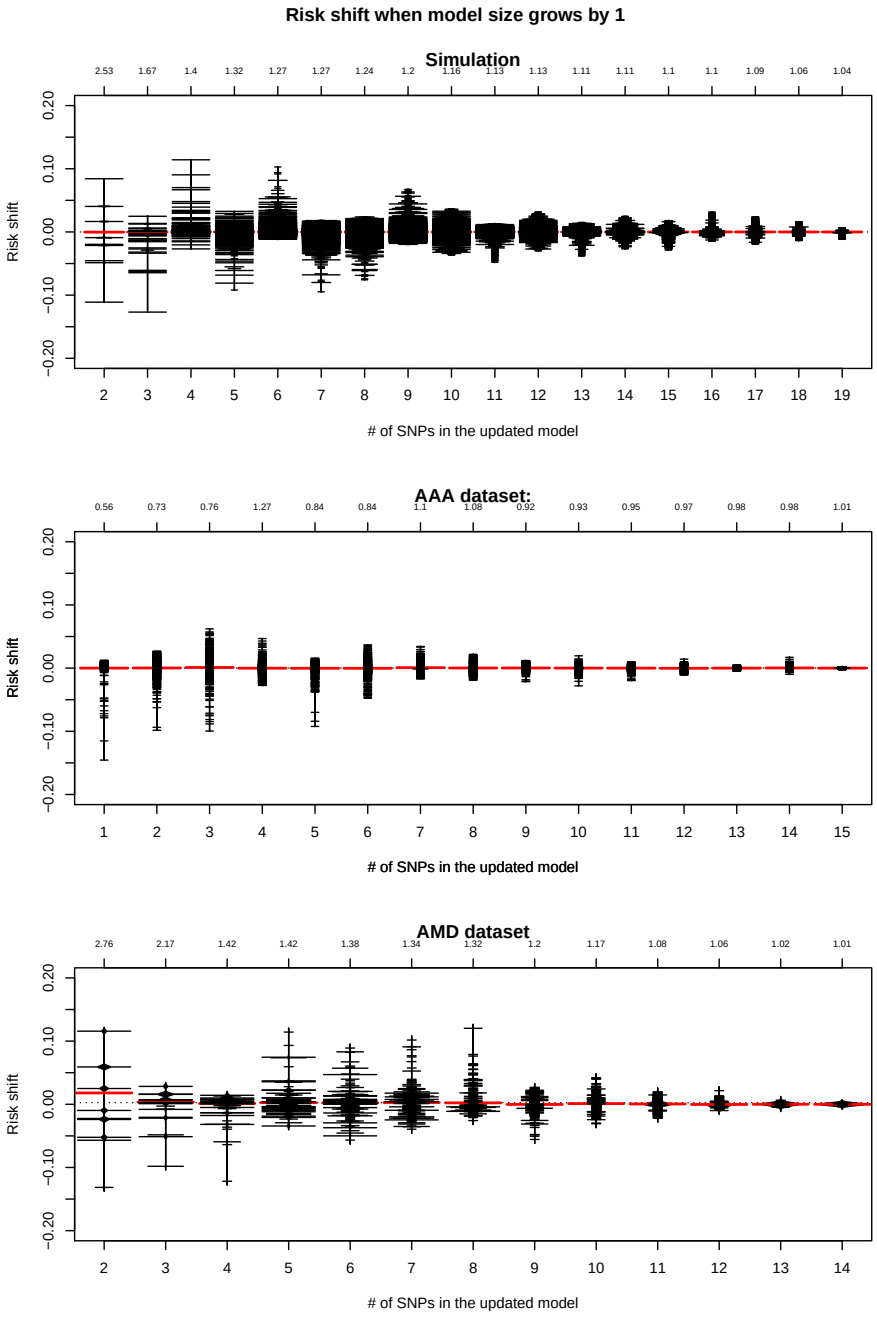


Figure 2.4: **The distribution of risk shifts as a function of the number of SNPs in the updated model.**

The plots were generated by the beanplot command in the R package of the same name [Kampstra, 2008]. The dark horizontal lines show individual observations. The width of the dark horizontal lines reveals the number of individuals sharing the same observation, with a fixed maximum boundary. The red lines indicate the means of the observations. The label above the plot is the added SNP's odds ratio in the model.

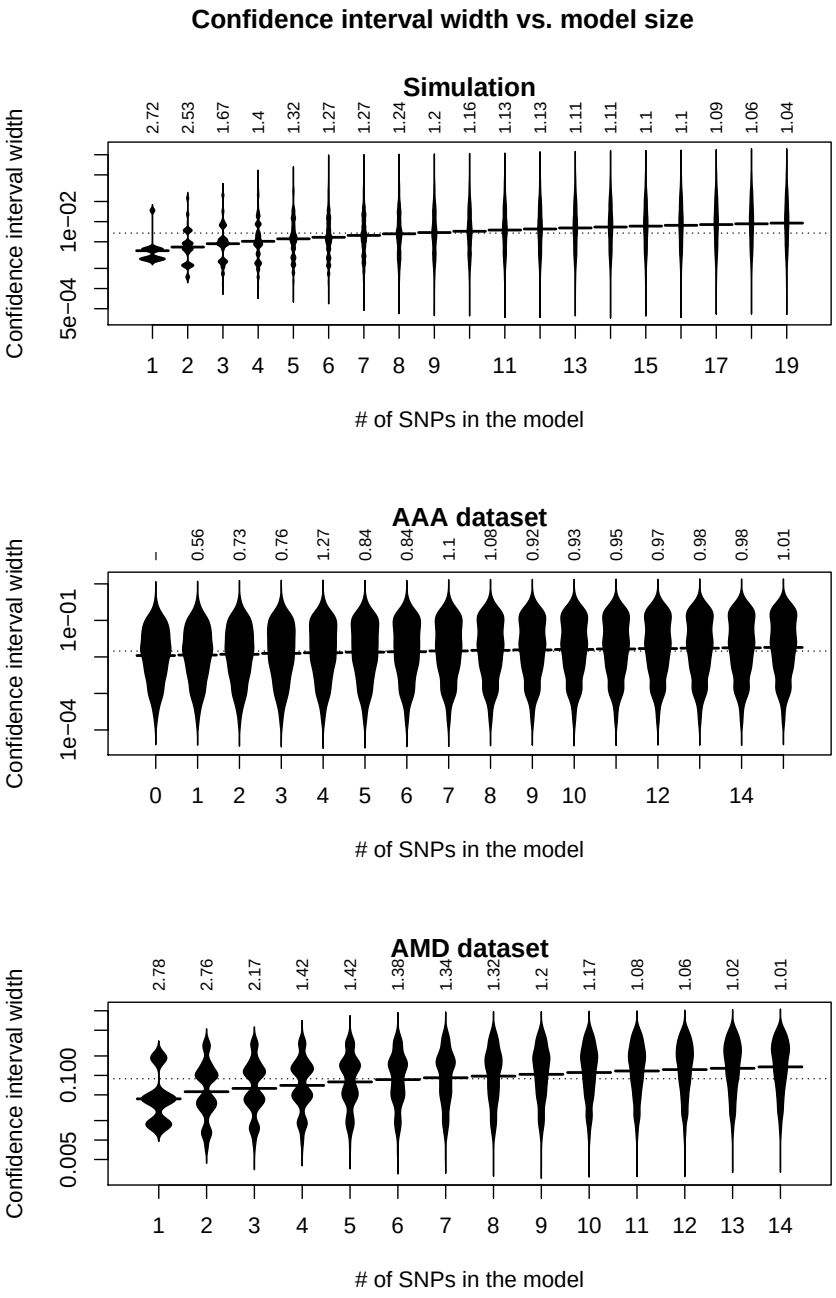


Figure 2.5: **The distribution of the confidence interval widths by model size.**

The confidence interval width axis uses the log scale. The label above the bean plot is the added SNP's odds ratio in the model. The horizontal line in the middle of each bean plot shows the mean value.

We then explored the influence of the model size on the confidence interval width. Figure 2.5 shows that the confidence interval width was positively correlated with model size in all the three data sets. For all the three data sets, the Spearman's rank correlation test gives p values smaller than 0.001, indicating strong positive correlation between the confidence interval width and model size. Table 2.4, 2.5, 2.6, and 2.7 also show the influence of the model size on the confidence interval width. More estimates have wider confidence intervals in the updated model with more SNPs than in the initial model with less SNPs. For the AAA data set, comparing the M_0 with M_8, 84.8% of the estimates have wider confidence intervals in the updated model compared to the initial model; while comparing the M_8 with M_{16}, 96.0% of the estimates have wider confidence intervals in the updated model compared to the initial model. For the AMD data set, comparing the M_1 to the M_{10}, 90.0% of the estimates have wider confidence intervals in the updated model compared to the initial model; while comparing the M_{10} to the M_{14}, 100% of the estimates have wider confidence intervals in the updated model compared to the initial model.

Furthermore, we determined the reclassification rates based on the screening strategies 2 and 3. The reclassification rates with bigger-effect SNPs in Table 2.4 and Table 2.6 are higher than that with smaller-effect SNPs in Table 2.5 and Table 2.7. But the small-effect SNPs can still affect the reclassifications. Table 2.5 shows that in the AAA data set, adding 8 less effective SNPs to the maxMRS-selected model, 19.0% of cases and 0.6% controls were correctly reclassified; while 0% of cases and 3.5% of controls were mistakenly reclassified. Table 2.7 shows that in AMD data set, adding 4 less effective SNPs to the maxMRS-selected model, 21.1% of the cases and 0% of the controls were correctly reclassified; while 0% of the cases and 13.0% of the controls were mistakenly reclassified. We also found the correctly reclassified rate of low*$\Leftrightarrow${-T,1] is much higher than [0,-T}$\Leftrightarrow${+T,1] for cases, and the correctly reclassified rate of low*$\Leftrightarrow${-T,1] is lower than [0,-T}$\Leftrightarrow${+T,1] for controls, in both the AAA data set and the AMD data set.

Table 2.4: **CI-augmented reclassification table for AAA data set, when the initial model only has the clinical predictors (M_0) and the updated model added 7 most effective SNPs (M_7).**

Outcome: Unaffected with AAA										
Initial Model: clinical predictors	Updated Model: clinical predictors +7 SNPs								[0, -0.033} ⇔ {+0.033,1]	LOW* ⇔ {-0.033,1]
	LOW* [0,0.033}		{-0.033}		{+0.033}		HIGH* {0.033,1]			
	-=	+	-=	+	-=	+	-=	+	%Reclassified	
LOW* [0,0.033}	99	323	0	19	0	1	0	0	2.1	4.5
{-0.033}	5	1	0	12	0	8	0	1		3.7
{+0.033}	0	0	5	7	0	9	0	4	11.9	
HIGH* {0.033,1]	0	0	0	4	3	10	1	92		

Outcome: Affected with AAA										
Initial Model: clinical predictors	Updated Model: clinical predictors +7 SNPs								[0, -0.033} ⇔ {+0.033,1]	LOW* ⇔ {-0.033,1]
	LOW* [0,0.033}		{-0.033}		{+0.033}		HIGH* {0.033,1]			
	-=	+	-=	+	-=	+	-=	+	%Reclassified	
LOW* [0,0.033}	2	17	0	4	0	0	0	0	13.9	17.4
{-0.033}	0	2	1	5	0	5	0	0		1.0
{+0.033}	0	0	1	4	0	8	0	3	4.3	
HIGH* {0.033,1]	0	0	1	2	2	9	6	152		

"LOW*"/"HIGH*" class records the number of samples with both risk estimates and the two confidence interval bounds lower/higher than the threshold, which is the prevalence of the corresponding disease. "-threshold" class records the number of samples with risk estimates lower than the threshold, but the higher confidence interval bounds above the threshold. "+threshold" class records the number of samples with risk estimates higher than the threshold, but the lower confidence interval bounds below the threshold. "% reclassified" is the percentage of samples that are reclassified from LOW*/HIGH* risk class to HIGH*/LOW* class. "-=" means the confidence interval width in the updated model is narrower than or equal to the width in the initial model. "+" means the confidence interval width in the updated model is wider than the initial width.

Table 2.5: **CI-augmented reclassification table for AAA data set, when the initial model has clinical predictors plus the 7 most effective SNPs (M_7) and the updated model has the clinical predictors plus all the 15 SNPs (M_{15}).**

Outcome: Unaffected with AAA										
Initial Model: clinical predictors + 7 SNPs	Updated Model: clinical predictors +15 SNPs								[0, -0.033} ⇔ {+0.033,1]	LOW* ⇔ {-0.033,1]
	LOW* [0,0.033}		{-0.033}		{+0.033}		HIGH* {0.033,1]			
	-=	+	-=	+	-=	+	-=	+	%Reclassified	
LOW* [0,0.033}	29	379	0	15	0	0	0	0	0.6	3.5
{-0.033}	0	1	3	40	0	3	0	0		0.6
{+0.033}	0	0	0	4	0	26	0	0	3.1	
HIGH* {0.033,1]	0	0	0	0	0	10	0	88		

Outcome: Affected with AAA										
Initial Model: clinical predictors + 7 SNPs	Updated Model: clinical predictors +15 SNPs								[0, -0.033} ⇔ {+0.033,1]	LOW* ⇔ {-0.033,1]
	LOW* [0,0.033}		{-0.033}		{+0.033}		HIGH* {0.033,1]			
	-=	+	-=	+	-=	+	-=	+	%Reclassified	
LOW* [0,0.033}	0	17	0	4	0	0	0	0	5.1	19.0
{-0.033}	0	0	1	15	0	2	0	0		0.0
{+0.033}	0	0	0	3	0	21	0	0	1.6	
HIGH* {0.033,1]	0	0	0	0	0	8	0	151		

"LOW*"/"HIGH*" class records the number of samples with both risk estimates and the two confidence interval bounds lower/higher than the threshold, which is the prevalence of the corresponding disease. "-threshold" class records the number of samples with risk estimates lower than the threshold, but the higher confidence interval bounds above the threshold. "+threshold" class records the number of samples with risk estimates higher than the threshold, but the lower confidence interval bounds below the threshold. "% reclassified" is the percentage of samples that are reclassified from LOW*/HIGH* risk class to HIGH*/LOW* class. "-=" means the confidence interval width in the updated model is narrower than or equal to the width in the initial model. "+" means the confidence interval width in the updated model is wider than the initial width.

Table 2.6: **CI-augmented reclassification table for AMD data set, when the initial model only has one SNP (M_1) and the updated model has 10 most effective SNPs (M_{10}).**

Outcome: Unaffected with AMD										
Initial Model: 1 SNP	Updated Model: 10 SNPs								[0, -0.055] ⇔ {+0.055,1]	LOW* ⇔ {-0.055,1]
	LOW* [0,0.055}		{-0.055}		{+0.055}		HIGH* {0.055,1]			
	-=	+	-=	+	-=	+	-=	+	%Reclassified	
LOW* [0,0.055}	8	12	0	5	0	6	0	1	21.9	37.5
{-0.055}	0	0	0	0	0	0	0	0		15.0
{+0.055}	0	0	0	0	0	0	0	0	55.0	
HIGH* {0.055,1]	3	0	1	7	0	5	0	3		

Outcome: Affected with AMD										
Initial Model: 1 SNP	Updated Model: 10 SNPs								[0, -0.055] ⇔ {+0.055,1]	LOW* ⇔ {-0.055,1]
	LOW* [0,0.055}		{-0.055}		{+0.055}		HIGH* {0.055,1]			
	-=	+	-=	+	-=	+	-=	+	%Reclassified	
LOW* [0,0.055}	6	28	0	21	0	23	0	4	32.9	58.5
{-0.055}	0	0	0	0	0	0	0	0		2.0
{+0.055}	0	0	0	0	0	0	0	0	15.6	
HIGH* {0.055,1]	4	0	1	27	8	38	2	125		

"LOW*"/"HIGH*" class records the number of samples with both risk estimates and the two confidence interval bounds lower/higher than the threshold, which is the prevalence of the corresponding disease. "-threshold" class records the number of samples with risk estimates lower than the threshold, but the higher confidence interval bounds above the threshold. "+threshold" class records the number of samples with risk estimates higher than the threshold, but the lower confidence interval bounds below the threshold. "% reclassified" is the percentage of samples that are reclassified from LOW*/HIGH* risk class to HIGH*/LOW* class. "-=" means the confidence interval width in the updated model is narrower than or equal to the width in the initial model. "+" means the confidence interval width in the updated model is wider than the initial width.

Table 2.7: **CI-augmented reclassification table for AMD data set, when the initial model has 10 most effective SNPs (M_{10}) and the updated model has the clinical predictors plus all the 14 SNPs (M_{14}).**

Outcome: Unaffected with AMD										
Initial Model: 10 SNP	Updated Model: 14 SNPs								[0, -0.055] ⇔ {+0.055,1]	LOW* ⇔ {-0.055,1]
	LOW* [0,0.055}		{-0.055}		{+0.055}		HIGH* {0.055,1]			
	-=	+	-=	+	-=	+	-=	+	%Reclassified	
LOW* [0,0.055}	0	20	0	3	0	0	0	0	0.0	13.0
{-0.055}	0	0	0	13	0	0	0	0		
{+0.055}	0	0	0	1	0	11	0	0	6.2	0.0
HIGH* {0.055,1]	0	0	0	0	0	1	0	3		

Outcome: Affected with AMD										
Initial Model: 10 SNP	Updated Model: 14 SNPs								[0, -0.055] ⇔ {+0.055,1]	LOW* ⇔ {-0.055,1]
	LOW* [0,0.055}		{-0.055}		{+0.055}		HIGH* {0.055,1]			
	-=	+	-=	+	-=	+	-=	+	%Reclassified	
LOW* [0,0.055}	0	30	0	8	0	0	0	0	2.3	21.1
{-0.055}	0	0	0	47	0	2	0	0		
{+0.055}	0	0	0	1	0	68	0	0	0.5	0.0
HIGH* {0.055,1]	0	0	0	0	0	23	0	108		

"LOW*"/"HIGH*" class records the number of samples with both risk estimates and the two confidence interval bounds lower/higher than the threshold, which is the prevalence of the corresponding disease. "-threshold" class records the number of samples with risk estimates lower than the threshold, but the higher confidence interval bounds above the threshold. "+threshold" class records the number of samples with risk estimates higher than the threshold, but the lower confidence interval bounds below the threshold. "% reclassified" is the percentage of samples that are reclassified from LOW*/HIGH* risk class to HIGH*/LOW* class. "-=" means the confidence interval width in the updated model is narrower than or equal to the width in the initial model. "+" means the confidence interval width in the updated model is wider than the initial width.

Finally, we evaluated the net benefit quantities of the three screening strategies. Table 2.8 shows that in both of the two data sets, the screening strategy of screening the individuals in the {-T,1] category provides the biggest net benefit quantity among the three strategies. The full models of both AAA and AMD data sets with {-T,1] screening strategy have the biggest net benefit quantity.

Table 2.8: **Net benefit of the classification of each model in the AAA data set and the AMD data set for the three screening strategies.**

	AAA			AMD		
Screening strategies	M_0	M_7	M_{15}	M_1	M_{10}	M_{14}
Screen individuals in {T,1]	0.201	0.191	0.180	0.601	0.386	0.318
Screen individuals in {+T,1]	0.220	0.218	0.216	0.601	0.587	0.590
Screen individuals in {-T,1]	0.236	0.238	0.242	0.601	0.730	0.753

2.4 DISCUSSION

Due to rapid progress and advancements in sequencing technology, it is now feasible, yet still expensive, to accurately type all genetic variants for an individual. To construct a risk estimate from these variants, we could attempt to use all of them or we could order them by estimated effect size, and use only the strongest predictors. But then the question is how many of these should be used. Clearly, as the effect size shrinks, adding a single small effect predictor to the risk model will not shift the risk by much. We explored here how the risk estimate and its certainty change as variants of decreasing effect size are added into the risk model, using simulated data and real data of two different complex diseases (AAA and AMD).

If we order SNPs by decreasing effect sizes and build risk models of various sizes by adding in the next SNP, we first observe that the risk shifts between successive models become more and more modest (Figure 2.2, Figure 2.4, Table 2.3) and the confidence intervals of the risk estimates tend to become larger (Figure 2.5, Table 2.4, Table 2.5, Table 2.6, and Table 2.7). Then, we observed that when the model size is large enough, if one more variant is added, the majority of the updated risk estimates will lie within the confidence interval of the preceding estimate and the confidence intervals of the new and old estimates will overlap substantially

(Figure 2.3). However, as we add multiple small-effect SNPs to the model simultaneously, these SNPs can still affect the reclassifications (Table 2.4, 2.5, 2.6, 2.7, and Table 2.8).

The large model with lots of SNPs are not appropriate for the disease risk prediction model, not only because the small effect SNPs increase the confidence interval of the prediction, while they do not shift the disease risk substantially, but also because the large prediction models may lead to higher genotyping cost. The higher genotyping cost is due to two causes: First, as the model size is larger, more SNPs need to be genotyped, which leads to a higher genotyping cost. Second, since no missing genotype data is allowed and bigger model has a bigger chance of having missing genotypes, more money has to be spent to fill in the missing data. Sometimes, another experiment has to be done. Blood may need to be redrawn and DNA may need to be re-extracted.

We recommend that all individuals with risk estimates above the threshold T or who have risk estimates with confidence intervals that overlap T (e.g., those in the {-T,1] category) should be screened, where the threshold T is chosen corresponding to the prevalence. There are two reasons for this. First, the strategy of screening the individuals in the {-T,1] category gives the biggest net benefit quantity among all three screening strategies. Second, for the cases, the correctly reclassified rate of low*$\Leftrightarrow${-T,1] is much higher than [0,-T}$\Leftrightarrow${+T,1], although for the controls, the correctly reclassified rate of low*$\Leftrightarrow${-T,1] is lower than [0,-T}$\Leftrightarrow${+T,1], in both the AAA data set and the AMD data set. Where screening costs much less compared to failing to detect the disease, screening the individuals in {-T,1] is the most appropriate strategy. However, it is important to remember that clinical cost-benefit analyses are complex and the assumption here is that screening is beneficial, although it is not necessarily so (for various diseases) if the "cost" of intervention risks are taken into account.

In the study, all the results are generated by one single split with 80% individuals in the training data set and 20% individuals in the testing data set. We then generated 5 more 80/20 random splits of the training and testing data sets to show the results change. Table 2.9 shows the maxMRS-selected models of each split. In the simulation data set, the maxMRS-selected models in the 5 testing data sets are similar; while in the AAA and AMD data sets, the maxMRS-selected models in the 5 testing data sets are variant. This is because

Table 2.9: **The number of SNPs in the maxMRS-selected models of five times 80/20 random splits in simulation data set, AAA data set, and AMD data set.**

Cross-validation	Simulation	AAA	AMD
1	16	8	10
2	16	8	11
3	17	9	12
4	17	13	-
5	17	14	-

The "-" symbol indicates that the maxMRS based on the full model is bigger than 0.06. The number of SNPs in the maxMRS-selected model of the five splits are sorted by an increasing order in each data set.

the sample size in the simulation data set is as large as 100,000, while the model sizes in the AAA and AMD data sets are as small as 2,626 and 1,015, respectively. Therefore, the max-MRS selected models should be built using data sets with large sample sizes. Otherwise, the max-MRS selected models may be affected greatly by the splitting of the training and testing data sets.

The main focus of the study is not providing accurate risks, but guiding the screening decision relative to a fixed threshold. So in the result section, we built models taking the risk shifts and the confidence interval width changes into account, but we did not take the true risk into account by evaluating whether the true risk is inside the confidence intervals. Here, we calculated the coverage probability of the 95% confidence intervals using the simulation data set (Figure 2.6). The maxMRS-selected model is 16 for simulated data set. When the model size is 16, the mean coverage probability is 0.64. When the model size increases 1, although both the confidence intervals and the risks do not shift much, it still possible that the confidence interval covers the true risk after the model size increases, but the confidence

interval does not cover the true risk before the model size increases. 2,414 individuals out of 20,000 have their confidence intervals covered the true risks using M_{19}, but do not have their confidence intervals covered the true risk using M_{18}. Two individuals have their confidence intervals covered the true risks using M_{18}, but do not have their confidence intervals covered the true risk using M_{19}. Although the coverage probabilities with maxMRS-selected models are not as high as we expected, the results do not conflict to our conclusion that the risks and the confidence intervals do not shift much when the model sizes exceed the maxMRS-selected models. Also, although the maxMRS-selected models may not be the best models for providing accurate risk estimates, they still can be the models that lead to the best screening strategies. If a true risk is very far from the threshold T, then even if the estimate of that risk is a bit inaccurate, this inaccuracy would be unlikely to alter the screening

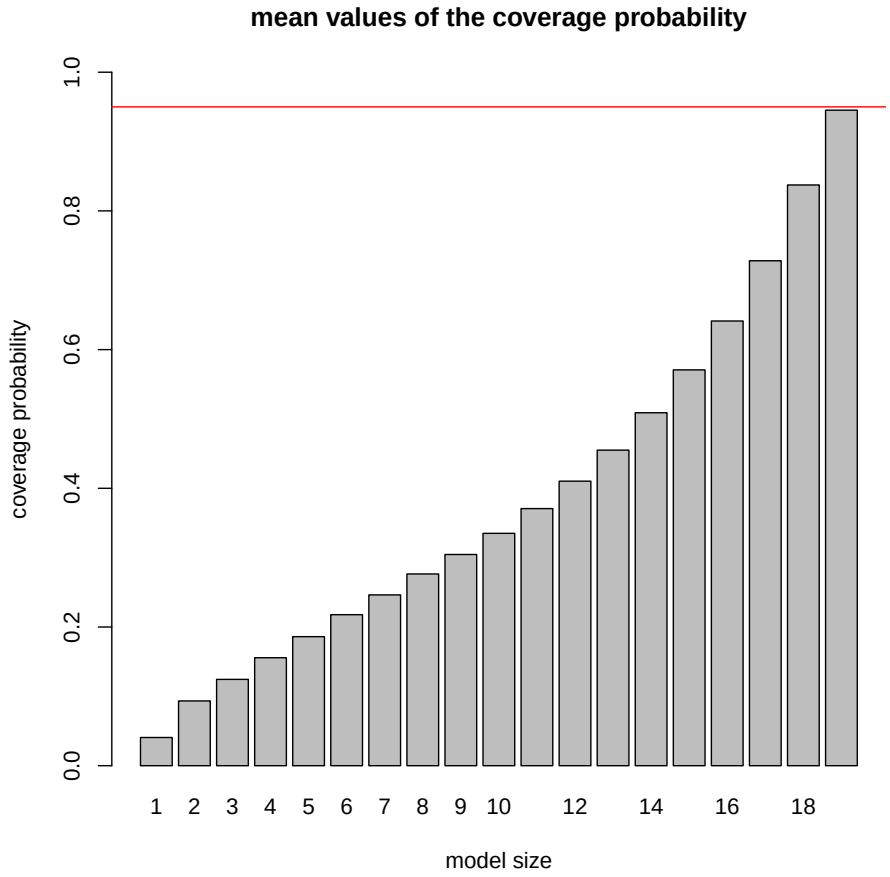


Figure 2.6: **Coverage probabilities of different model sizes in the simulation data set.**

In our results, the relationship of the risks and the confidence interval widths is consistent with the binomial distribution property that the confidence interval width increases as the risk estimate rises to 0.50 and decreases as the risk estimate increases beyond 0.5. Since the disease prevalence in the simulation study, AAA study, and AMD study were 0.055, 0.033 and 0.055, respectively, most of the risk estimates were much lower than 0.5, in all three data sets. In the simulation data set, AAA data set and AMD data set, only three, one, and eight individuals had risk estimates bigger than 0.5, respectively. In all the three data sets, most of the confidence intervals increased as the risk increased, or decreased as the risk decreased, when one more SNP with the next largest effect size was added to the model. But there

were still some confidence intervals that increased as the risks decreased in the three data sets and some confidence intervals that decreased as the risks increased in the AAA data set only. These two scenarios are because of two reasons. The first one is that the confidence interval widths are not only related to the risk size, but are also related to the model size. Even though the risks estimated by larger models are smaller, the confidence intervals can still become bigger if the model sizes are bigger. The second reason is that when the risk estimate exceeds 0.50, the confidence interval width decreases as the risk increases, and vice versa.

The risk trajectory plot (Figure 2.2) shows that the higher-initial-risk individuals have their risks shifted more than the lower-initial-risk individuals as more SNPs are added to the model. This observation is mainly because of two reasons. First, the risk trajectories that start with a low initial risk suffer from a lower bound effect - they can not move very far in the down direction. Second, since the disease prevalence in the three data sets is as low as 0.055, 0.033 and 0.055, respectively, the majority of people should be in the low risk category. If the risks in the low category rise up substantially, then the prevalence would exceed its expectation.

Other previous studies classified individuals using both the risks and the confidence intervals. Goddard and Lewis [2010] developed a strategy, which has been implemented in the R package REGENT [Crouch et al., 2013], to classify individuals into risk classes using the risk and the confidence interval of an average individual to anchor the classification. With N SNPs, there are 3^N genotypes. The "average individual" is the individual with a genotype relative risk closest to the average risk, which is the sum across all the 3^N genotypes of the products of their frequencies and relative risks of disease. An estimate with confidence interval overlapping the confidence interval of the "average individual" is classified as "Average" risk. An estimate with confidence interval below the confidence interval of the "average individual" is categorized as "low" risk. In a similar manner, they also define "moderate" and "high" risk categories. Scott et al. [2013] applied the reclassification method and the REGENT R package to predict the risk of rheumatoid arthritis and its age of onset with smoking. In Goddard and Lewis [2010] , they observed that when one uses confidence interval-based risk classification, one can run into the situation where an individual with a

lower risk is classified into the high risk group because their confidence interval was larger than an individual with a slightly higher risk who had a narrower confidence interval. This phenomenon also happens in our AAA and AMD data sets. We recorded the smallest risk estimate among those whose upper bounds of the confidence intervals are higher than the threshold. Then, we counted the number of estimates that are higher than this smallest risk estimate, but with confidence intervals that do not cross the threshold. Using the smallest model (model 1) and the biggest model (model 16) of the AAA data set, models 1, 11, and 14 of the AMD data set, there are 12, 24, 0, 19 and 9 estimates that meet these criteria, respectively.

Hart et al. [2013] also built a logistic regression model for risk estimation and took confidence intervals into account. They used logistic regression to create a new actuarial risk assessment instrument (ARAI). They categorized the individuals to two groups based on the ARAI score. They evaluate the ARAI at both group level and the individual level. Their results at the individual level are similar to our results. The mean width of the 95% confidence intervals for individual risk estimates in the high risk score category was much bigger than that of subjects in the low risk category. Confidence intervals for individual risk estimates overlapped completely within groups, and almost completely across groups.

Consideration of risk estimate uncertainty is important because if the disease risk estimates, as well as the confidence intervals are provided, people can make more informed decisions regarding their screening decisions [Weeks and Ott, 1990]. For example, suppose an individual has a risk estimate below the threshold, but the upper bound of the confidence interval is much higher than the threshold. If only the risk estimate is provided, there will be an unfounded confidence in the estimate and the individual may feel safe, and therefore may choose to not undergo screening. But if both the risk estimate and its confidence interval are provided, the individual may no longer feel safe, and probably will undergo screening. For another example, consider an individual with a risk estimate slightly higher than the threshold and the lower bound of the confidence interval also above the threshold. If only the risk estimate is provided, this individual may not undergo screening, because the risk estimate is only slightly higher than the threshold. However, if the confidence interval shows that it has 95% certainty that the individual has high risk of getting the disease, then this

individual may decide to undergo screening. On the other hand, since it is difficult to clearly convey risk estimates in such a way that they are understood and interpreted correctly, it may be even more difficult to clearly communicate the information embodied in the confidence intervals around those risk estimates [Lautenbach et al., 2013]. Careful consideration of how to best communicate these measures of risk estimate uncertainty is merited, lest such communications lead to increased disease-related anxieties and poorer risk perceptions [Han et al., 2011; Han, 2013].

3.0 AIM 2. A BAYESIAN APPROACH FOR SNP EFFECT ESTIMATION AND GENETIC RISK PREDICTION FOR DICHOTOMOUS TRAITS

3.1 BACKGROUND AND SIGNIFICANCE

Sequencing techniques have been developing quickly in recent years. They help people to explore our genetics and to learn diseases deeply at the genotype level. Whole genome sequencing is one of the techniques, which is of low price and is widely used in genetic research. SNPs are the most common type of genetic variants, which can occur in both coding regions and non-coding regions of the genome. The SNPs in the coding regions can alter amino acid coding and thus be considered functional and can cause diseases. SNPs in non-coding regions can also cause disease. Whole-genome microarrays can detect over 4 million markers per sample, and next-generation sequencing (NGS)-based whole-genome sequencing provides a base-by-base method for detecting the 3.2 billion bases of the human genome. SNPs are usually used to study all kinds of phenotypes. But only a few SNPs are effective and predictive of a given phenotype, while most of the SNPs have no effect or only small effects on that phenotype. Estimating the SNP effects helps to study those phenotypes and predict the risk of getting the phenotypes.

Different study designs and different research questions lead to different data types. Data with dichotomous traits is one of the most common data types. A dichotomous trait has only two phenotype statuses: affected or not affected. When analyzing quantitative traits, the methods are based on linear regression models. When analyzing dichotomous traits, the methods are based on logistic regression models.

There are several existing methods for estimating SNP effects. GWAS is one of the most widely used methods detecting the disease associated SNPs. In GWAS, regression is

conducted on each SNP, as well as relevant covariates. GWAS is easy to conduct for data with both quantitative traits and the dichotomous traits. Therefore, a great many GWAS and GWAS based meta-analysis have been conducted, since the first GWAS was applied in 2005. However, the drawback of this method is that when evaluating the effect of an SNP, it usually fails to account for the effects of other SNPs, which decreases the precision and the power to detect true associations [Moser et al., 2015].

To predict the risk of getting a disease, people build a genetic risk model with a subset of SNPs. In GWAS, these SNPs are selected based on the corresponding p values that are below the multiple-comparison threshold. There are other methods detecting and selecting the predictive SNPs. The features of sparsity and shrinkage of regression coefficients of the least absolute shrinkage and selection operator (LASSO) method are used for SNP selection [Feng et al., 2012].

Erbe et al. [2012] proposed a Bayesian mixture model called BayesR, which selects the predictive SNPs and predicts outcomes. In the MCMC process, Erbe et al. assigned a known value to the genetic variance. Later on, Moser et al. [2015] changed the BayesR a little bit so that their genetic variance is informed from the data. BayesR enables the simultaneous fitting of all the SNPs. It treats the SNP effects as drawn from a prior distribution of a mixture of normal distributions with mean of zero but different variances. Different variances in the mixture normal distribution allow the SNP effect prior distribution approaches the true distribution of the effect sizes. Sparseness is accomplished by setting a zero effect category. However, BayesR method is designed be applied to data with quantitative traits. When applying BayesR to a dichotomous trait outcome data set, Moser et al. treats the binary outcome coded 0/1 as the response in an ordinary linear regression. This may cause the following problems: 1) when fitting an ordinary linear model, the residuals do not follow normal distributions, but a standard logistic distribution, which may cause some bias of the SNP effect estimation and the risk prediction; 2) the SNP odds ratios cannot be calculated using the estimated SNP effects; 3) it is hard to explain the predicted outcome: the predicted phenotypes can be bigger than 1, so it cannot be treated as the probability of being a case. Our extension of BayesR based on Erbe et al. and Moser et al. (BayesRB without and with genetic variance fixed, respectively) is a better method for analyzing dichotomous traits.

This new method can be used to select associated SNPs, estimate the SNP effects and make the prediction of the disease risks, taking account for the effects of all the genotyped SNPs.

Several previous studies have proposed methods on dealing with dichotomous outcome data. Directly fitting of a Bayesian logistic model is a method that can be used. However, one drawback of fitting a Bayesian logistic model directly is having no conjugate prior. Most previous approaches applied Metropolis-Hastings, or otherwise used accept-reject steps [Chen and Dey, 1998; Gamerman, 1997]. These make each MCMC step complicated and consumes much computational time. When hundreds of thousands of SNPs are there in the data sets, it is necessary that each MCMC step runs quickly. Simple Gibbs sampling can be used when the auxiliary variable models are used. Albert and Chib [1993] proposed an approximate data-augmentation algorithm. They used the $t(8)$ quantile to approximate the logistic quantiles. The posterior distributions of all the parameters have standard forms. Thus, Gibbs sampling can be applied. But this is an approximate method, instead of an exact one. Polson et al. [2013] proposed another data-augmentation strategy for Bayesian logistic regression, which is a direct analogue of Albert and Chib construction. Their approach is based on a newly proposed Polya-Gamma distribution family. But it is not practical in the Bayesian mixture model. Holmes and Held [2006] proposed an exact data-augmentation algorithm. One posterior distribution does not have a standard form. They used rejection sampling to solve the issue. However, adaptive-rejection sampling only updates individual coefficients, which leads to a poor mixing when coefficients are correlated. Therefore, they suggested a joint updating of some parameters whose posteriors correlated to each other. This allows fast mixing in the chain.

3.2 METHODS

3.2.1 BayesRB Approach

To estimate each SNP effect taking account for all the SNP effects, we constructed a Bayesian mixture model and assumed the SNP effects come from mixtures of several normal distributions, including one with zero mean and zero variance. We applied the Markov chain Monte Carlo (MCMC) to estimate the unknown parameters. We used a Gibbs scheme and Metropolis-Hasting scheme to sample values from each unknown parameter's conditional posterior distribution. Then, we made inference of each unknown parameters, including the effect sizes of the SNPs and the categories the SNPs belongs to.

In a sample with n independent individuals and p independent SNPs, the phenotypes are related to SNPs with a logistic regression model

$$\log\left(\frac{P(y=1)}{1-P(y=1)}\right) = 1_n\mu + X\boldsymbol{\beta}, \tag{3.1}$$

where y is an n-dimensional vector of dichotomous phenotypes, $P(y=1)$ is the probability of being affected, 1_n is an n-dimensional vector of ones, μ is the general mean of the $P(y=1)$ in the logit scale, X is a $n \times p$ matrix of genotypes coded as 0, 1, or 2 indicating 0, 1, or 2 risk alleles. The vector $\boldsymbol{\beta}$ is a p-dimensional vector of SNP effects.

To extend the BayesR approach to binary traits, we introduced an auxiliary variable Z_i.

$$Z_i = \mu + \sum_{j=1}^{p} X_{ij}\beta_j + \epsilon_i, \tag{3.2}$$

where ϵ_i follows a standard logistic distribution. For the SNP j, $\boldsymbol{X_{.j}}$ is standardized with the mean of 0 and the variance of 1, where $\boldsymbol{X_{.j}}$ is the vector of the number of risk alleles of SNP j for all the individuals. To keep the conditional conjugacy for updating β, we introduced a further set of variables, $\boldsymbol{\lambda}$, which contains λ_i, $i = 1, ..., n$. Then, the model becomes

$$y_i \sim \begin{cases} 1 & Z_i > 0 \\ 0 & otherwise \end{cases}$$

$$Z_i = \mu + \sum_{j=1}^{p} X_{ij}\beta_j + \epsilon_i$$

$$\epsilon_i \sim N(0, \lambda_i)$$

$$\lambda_i = (2\psi_i)^2$$

$$\psi_i \sim KS, \tag{3.3}$$

where $\psi_i, i = 1, ..., n$, are independent variables following the Kolmogorov-Smirnov (KS) distribution. This model is equivalent to the logistic regression model [Holmes and Held, 2006]. The SNP_j's effect β_j is assumed to be a mixture of four zero mean normal distributions.

$$Pr(\beta_j|\pi, \sigma_g) \sim \sum_{k=1}^{4} \pi_k N(0, C_k\sigma_g^2), \tag{3.4}$$

where $k = (1, 2, 3, 4)$. $(C_1, C_2, C_3, C_4) = (0, 10^{-4}, 10^{-3}, 10^{-2})$. σ_g^2 is the genetic variance. There are two strategies to deal with σ_g^2, according to Erbe et al. and Moser et al., respectively: 1) treating σ_g^2 as fixed; 2) treating σ_g^2 as random. When treating σ_g^2 as random, we set σ_g^2 follows a uniform non-informative prior with the initial value follows $N(0, 200)$. $\boldsymbol{\pi} = (\pi_1, \pi_2, \pi_3, \pi_4)$ are the mixture proportions, which sum up to 1. The prior for π is a symmetric Dirichlet distribution:

$$Pr(\pi_1, \pi_2, \pi_3, \pi_4) \sim D(\delta, \delta, \delta, \delta) \tag{3.5}$$

with $\delta = 1$.

For the MCMC process, the fully conditional posterior distributions of each unknown parameters are given below. The proof can be found in the appendix. We use $|.$ to represent conditioned on the data and all other parameters. The dependency diagram treating σ_g^2 as fixed can be found in Figure 3.1, and treating σ_g^2 as random can be found in Figure 3.2.

1. Update $\{\boldsymbol{Z}$ and $\lambda\}$ jointly from their joint conditional posterior distribution:

$$P(\boldsymbol{Z}, \boldsymbol{\lambda}|.) = P(\boldsymbol{\lambda}|\boldsymbol{\beta}, \mu, \boldsymbol{X}, \boldsymbol{Z})P(\boldsymbol{Z}|\boldsymbol{\beta}, \mu, \boldsymbol{Y}, \boldsymbol{X}) \tag{3.6}$$

In this case, the $\boldsymbol{Z}|\boldsymbol{\beta}, \mu, \boldsymbol{Y}, \boldsymbol{X}$ follows independent truncated logistic distribution as shown below:

$$\boldsymbol{Z}|\boldsymbol{\beta}, \mu, \boldsymbol{Y}, \boldsymbol{X} \propto \begin{cases} Logistic(\mu + \sum_{j=1}^{p} X_{ij}\beta_j, 1)I(Z_i > 0) & if\ y_i = 1 \\ Logistic(\mu + \sum_{j=1}^{p} X_{ij}\beta_j, 1)I(Z_i \leq 0) & otherwise, \end{cases} \tag{3.7}$$

where $Logistic(\theta, \kappa)$ is the density function of the logistic distribution with a mean of θ and a scale of κ.

The conditional distribution $P(\boldsymbol{\lambda}|\boldsymbol{\beta}, \mu, \boldsymbol{X}, \boldsymbol{Z})$ does not have a standard form. Therefore, we used a rejection sampling process to sample the λ's.

2. The conditional posterior distribution of the general mean μ is

$$\mu|. \sim N(n^{-1}\sum_{i=1}^{n}(Z_i - \sum_{j=1}^{p} X_{ij}\beta_j), \frac{\sum_{i=1}^{n}\lambda_i}{n^2}) \tag{3.8}$$

μ is sampled from the above distribution for each cycle of the Markov chain.

3. For SNP j, β_j and b_j are updated jointly from their joint conditional posterior distribution, where $\boldsymbol{b}$ denotes the category that each SNP belongs to. SNP j belongs to cetegory k (k=1, 2, 3, or 4): $b_j = k$.

$P(b_j, \beta_j|.)$

$$\propto P(\beta_j|b_j, \boldsymbol{Z}, X, \sigma_g^2, \boldsymbol{\lambda}, \mu, \boldsymbol{\beta_{-j}}, \boldsymbol{b_{-j}}, \boldsymbol{\pi})P(b_j|\boldsymbol{Z}, X, \sigma_g^2, \boldsymbol{\lambda}, \mu, \boldsymbol{\beta_{-j}}, \boldsymbol{b_{-j}}, \boldsymbol{\pi}), \tag{3.9}$$

where $\boldsymbol{b_{-j}}$ denotes the vector of categories that all the SNPs expect SNP j belong to. $\boldsymbol{\beta_{-j}}$ denotes the vector of effects of all the SNPs expect SNP j. $\boldsymbol{C}$ is the vector (0, 10^{-4}, 10^{-3}, 10^{-2}). Then,

$$C_{b_j} \sim \begin{cases} 0 & b_j = 1 \\ 10^{-4} & b_j = 2 \\ 10^{-3} & b_j = 3 \\ 10^{-2} & b_j = 4 \end{cases}$$

.

For each SNP, b_j is updated first. Set $T_k = P(b_j = k|X, \boldsymbol{Z}, \sigma_g^2, \boldsymbol{\lambda}, \mu, \boldsymbol{\beta_{-j}}, \boldsymbol{b_{-j}}, \boldsymbol{\pi})$.

$$T_k = P(b_j = k|X, \boldsymbol{Z}, \sigma_g^2, \boldsymbol{\lambda}, \mu, \boldsymbol{\beta_{-j}}, \boldsymbol{b_{-j}}, \boldsymbol{\pi}) = \frac{1}{\sum_{l=1}^{4} exp(logL_{jl} - logL_{jk})}, \tag{3.10}$$

where k is the category SNP j is assigned to. $k = (1, 2, 3, 4)$.

If $k \neq 1$,

$logL_{jk} = -\frac{1}{2}\left\{log(\prod_{i=1}^{n}\lambda_i) + log(\sum_{i=1}^{n}\frac{X_{ij}^2}{\lambda_i}C_{b_j}\sigma_g^2 + 1)\right\}$

$$-\frac{1}{2}\Bigg[-\frac{(\sum_{i=1}^{n}\frac{\tilde{Z_{ij}}X_{ij}}{\lambda_i})^2}{\sum_{i=1}^{n}\frac{X_{ij}^2}{\lambda_i} + \frac{1}{C_{b_j}\sigma_g^2}} + \sum_{i=1}^{n}\frac{\tilde{Z_{ij}}^2}{\lambda_i}\Bigg] + log(\pi_k). \tag{3.11}$$

If $k = 1$,

$$logL_{j1} = -\frac{1}{2}\left\{log(\prod_{i=1}^{n}\lambda_i)\right\} - \frac{1}{2}\Bigg[\sum_{i=1}^{n}\frac{\tilde{Z_{ij}}^2}{\lambda_i}\Bigg] + log(\pi_1). \tag{3.12}$$

The SNP j is assigned to category k based on a value h sampled from a uniform distribution.

$$b_j = \begin{cases} 1 & if\ 0 < h \leq T_1 \\ 2 & if\ T_1 < h \leq T_1 + T_2 \\ 3 & if\ T_1 + T_2 < h \leq T_1 + T_2 + T_3 \\ 4 & if\ T_1 + T_2 + T_3 < h \leq 1. \end{cases} \tag{3.13}$$

Then, we updated β_j:

$$\beta_j|b_j, \boldsymbol{Z}, X, \sigma_g^2, \boldsymbol{\lambda}, \mu, \boldsymbol{\beta_{-j}}, \boldsymbol{b_{-j}}, \boldsymbol{\pi} \sim \begin{cases} \delta(\beta_j) & b_j = 1 \\ N(\frac{\sum_{i=1}^{n}\tilde{Z_{ij}}X_{ij}\lambda_i^{-1}}{\sum_{i=1}^{n}\lambda_i^{-1}X_{ij}^2 + \frac{1}{C_{b_j}\sigma_g^2}}, \frac{1}{\sum_{i=1}^{n}\lambda_i^{-1}X_{ij}^2 + \frac{1}{C_{b_j}\sigma_g^2}}) & b_j \neq 1 \end{cases}, \tag{3.14}$$

where $\tilde{Z_{ij}} = Z_i - \mu - \sum_{l \neq j} X_{il}\beta_l$, and $\delta(\beta_j)$ denotes the dirac delta function with all probability mass at $\beta_j = 0$ if $b_j = 1$.

For each cycle of the Markov chain, b_j and β_j for SNP j are updated using the above distribution. Then, we repeated step 3 for SNP $j + 1$, ..., p, and recorded the number of SNPs being in each category as m_k. $k = (1, 2, 3, 4)$.

4. If treating σ_g^2 as fixed, then skip this step. If estimating the σ_g^2, then we set a uniform prior for σ_g^2 and updated it using the Metropolis-Hasting sampling for each cycle of the

Markov chain. We set the truncated normal distribution with variance of θ as the proposal function. The detail can be found in the Appendix.

5. The conditional posterior distribution of the mixture proportion π follows

$$\boldsymbol{\pi}|. \sim D(m_1 + \delta, m_2 + \delta, m_3 + \delta, m_4 + \delta), \tag{3.15}$$

where $\delta = 1$. And $\boldsymbol{\pi}$ is sampled from the above distribution for each cycle of the Markov chain.

6. Following Moser et al. [2015], in order to increase mixing, we randomly permuted the SNP orders. Then, loop back to step 1.

The pseudo code of the above MCMC steps can be found in the appendix.

After the MCMC steps, the parameters can be estimated by calculating the means of the sampled values from their posterior probabilities. We also recorded the proportion of iterations that each SNP was assigned to category 2, 3 or 4. If the proportion exceeds the threshold, the SNP is the BayesRB selected associated SNPs.

Then, we made risk prediction of being affected for the individuals in the testing data set. We set $W_{ij} = (X_{ij} - 2p_j)/\sqrt{2p_j(1-p_j)}$. X_{ij} is the number of copies of the risk alleles (0, 1, 2) at SNP j for individual i with p_j being the frequency of the risk allele in the training population.

$$\hat{Z}_i = \hat{\mu} + \sum_{\hat{\beta}_j > 0} W_{ij}\hat{\beta}_j \tag{3.16}$$

$$\hat{P}(Y_i = 1) = ilogit(\hat{Z}_i), \tag{3.17}$$

where $\hat{\mu}$ and $\hat{\beta}_j$ are estimated from the MCMC above. $\hat{P}(Y_i = 1)$ is the estimated probability of being affected for a new individual in the testing data set.

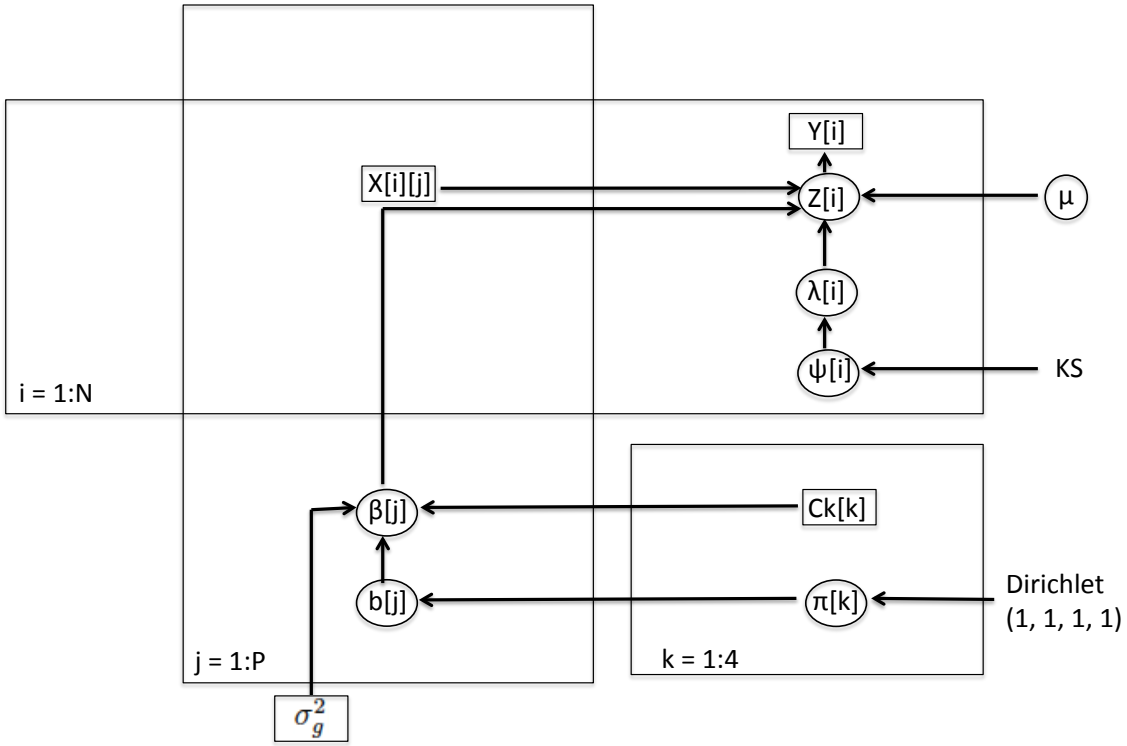


Figure 3.1: **Dependency diagram of the parameters if treating σ_g^2 as fixed.**

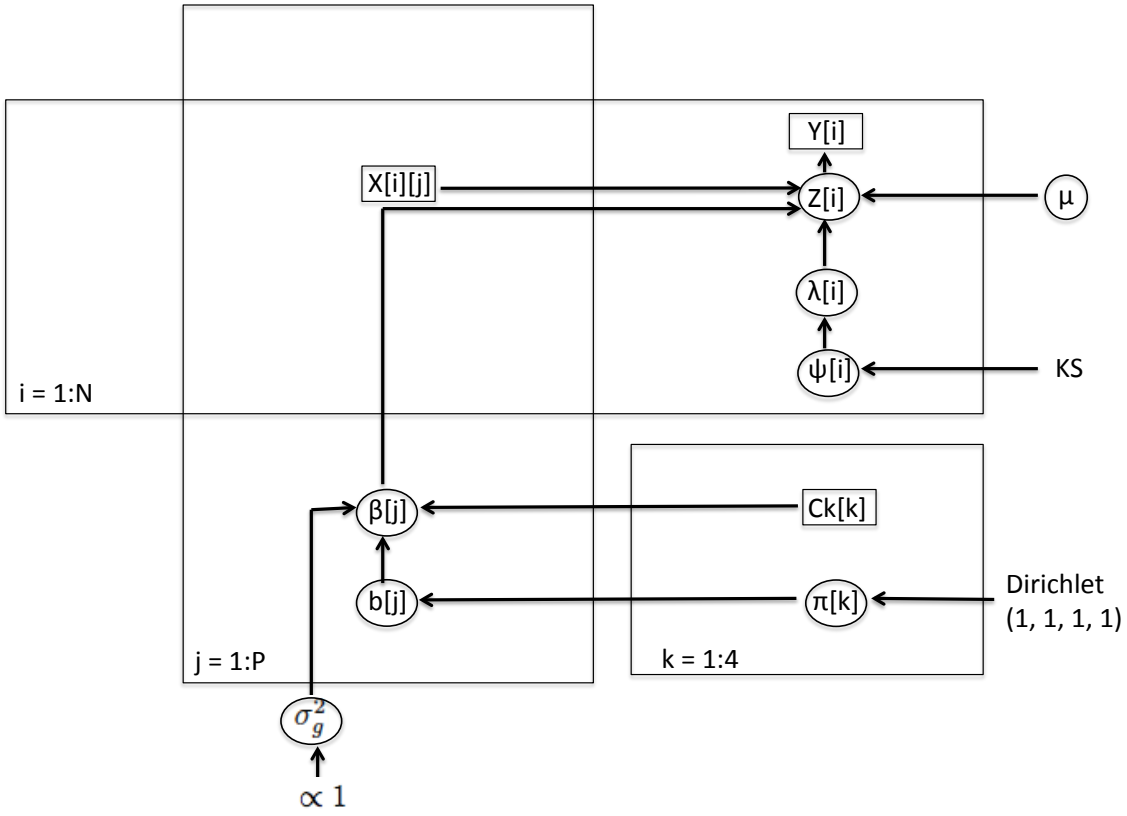


Figure 3.2: **Dependency diagram of the parameters if treating σ_g^2 as random.**

3.2.2 Other Approaches

BayesR

BayesR Fortran program [Moser et al., 2015] generates the SNP effect estimates, predicted outcomes, and the proportion of iterations that each SNP is assigned to each category. BayesR SNP effect estimates and the predicted outcomes are directly obtained from the output. The BayesR selected associated SNPs are those with the proportions of iterations that the SNPs are assigned to category 3 or 4 bigger than the threshold (Table 3.1).

Logistic Regression

We used PLINK [Purcell et al., 2007] to conduct all the logistic regression analysis. The SNP effects are estimated by the marginal logistic regression method. The logistic regression selected associated SNPs are those with p values smaller than the Bonferroni-corrected threshold. Among the logistic regression selected associated SNPs, we selected one SNP with the smallest p value in each linkage disequilibrium (LD) block to do the prediction.

LASSO

We also used PLINK [Purcell et al., 2007] to conduct all the LASSO analysis. The LASSO selected associated SNPs are those whose SNP effects are not shrunk to zero with the shrinkage parameter λ. The SNP effects for the LASSO selected associated SNPs are estimated using marginal logistic regression. The other SNP effects are zero. The predicted disease risks are estimated by the logistic regression using all the LASSO selected associated SNPs.

Table 3.1: **An overview of how the BayesRB, BayesR, logistic regression and LASSO results are generated.**

Approaches	**Criteria to select associated SNPs**	**Associated SNPs**	**SNP Effects**	**Risk Prediction**
BayesRB	Proportion of iterations that a SNP is assigned to category 2, 3 or 4	SNPs with the proportion bigger than the threshold	Generated by bayesRB algorithm	Using all the SNP effect estimates
BayesR	Proportion of iterations that a SNP is assigned to category 3 or 4	SNPs with the proportion bigger than the threshold	Generated by bayesR algorithm	Using all the SNP effect estimates
Logistic Regression	Bonferroni-corrected P values	SNPs have the p values smaller than the threshold	Generated by marginal logistic regression	Among the selected SNPs, using uncorrelated SNPs with smallest p values
LASSO	SNPs that are not shrunk to 0		None-zero effect SNP estimates are re-estimated by logistic regression	Using all the selected SNPs

	Pilot Simulated Data Sets		Genome-wide Simulated Data Sets	Real Data Sets (WTCCC)	
Details	Unrealistic	Realistic	80/20 training/testing, 50 replicates	CD	BD
				80/20 training/ testing, 20 replicates	
# of individuals	5000	5000	3000	4358	4445
# of SNPs	3000	3000	~260,000	312,035	306,702
Minor Allele Frequency	>0.1	>0.1	>0.1	>0.1	>0.1
Simulated SNP Effects	Match the bayesRB β prior. β 's are bigger than reality.	Match the β distribution of Crohn's disease in Wu et al. [2013]	195 causal SNPs match the β distribution in Wu et al. [2013]	-	
Purpose	1. Verify that the bayesRB program works properly. 2. detect the random/fixed genetic variance		Evaluate the bayesRB performance when SNPs are not fully independent.	Evaluate the bayesRB performance in real data sets.	
SNP Effect Estimation	Compare the bayeRB effect estimates to bayesR and logistic regression				
Associated SNP Selection	-		Compare the bayeRB associated SNP selection performance to bayesR, lasso, and logistic regression		
Risk Prediction	-	-	Compare the bayeRB risk prediction performance to bayesR, lasso, and logistic regression		

Figure 3.3: **Pilot simulated data sets, genome-wide simulated data sets, and real data sets description and the study design diagram of Aim 2.**

3.2.3 Study Design

In this study, we proposed a Bayesian approach (BayesRB) to select the associated SNPs, estimate the SNP effects and predict genetic risk for dichotomous traits. We applied the approach to pilot simulated data sets, genome-wide simulated data sets and real data sets. There are two pilot simulated data sets. One has unrealistic SNP effects, while the other has more realistic SNP effects. Using the pilot simulated data sets, we diagnosed the performance of the BayesRB approach. To examine whether the parameters mix well, we assessed the convergence of the parameters. We also explored whether it is necessary to infer the genetic

variance from the data and how genetic variance affects the SNP effect estimation. We diagnosed the performance of the BayesRB approach with and without genetic variance fixed. The genome-wide simulated data set is even more like a real data set with some SNPs correlated to each other, showing realistic linkage disequilibrium patterns. Using the genome-wide simulated data set, we aimed to measure BayesRB's performance when the SNPs are not fully independent of each other. Considering the computational time, we simulated 50 genome-wide simulated data sets. Each has 3000 individuals. For each one, we randomly selected 80% of individuals to the training data set, and the rest to the testing data set, letting equivalent proportions of the cases to the controls in the two data sets. We selected the associated SNPs and estimated the SNP effects using the training data set and predicted the risk of being affected using the testing data set. We used the Crohn's disease (CD) and bipolar disorder (BD) data set from Welcome Trust Case Control Consortium (WTCCC) study as the real data sets to investigate the BayesRB approach. We created 20 random 80/20 training and testing splits of the two data sets. The purpose of using these two WTCCC data sets is to investigate the performance of BayesRB in real data sets.

The data description and the study design diagram of aim 2 can be found in Figure 3.3.

3.3 RESULTS

3.3.1 BayesRB R package

We wrote a BayesRB R package using RCPP. All the following results are generated by the BayesRB R package. The source code can be found in the following website:
https://github.com/sylviashanboo/BayesRB

3.3.2 Pilot Simulated data sets

To ensure the program works correctly, we first simulated two pilot simulated data sets. In both of the data sets, the SNPs are independent and the individuals are independent.

We simulated both data sets with 5,000 individuals and 3,000 SNPs using the Multiple Gene Risk Prediction Performance (mgrp) R package [Pepe et al., 2010b]. One data set has unrealistic SNPs effects. The SNP effects were set bigger than those usually seen in a typical GWAS and matching the prior of $\boldsymbol{\beta}$ as Moser et al. [2015] did: 50 SNP effects have variance of $10^{-2}\sigma_g^2$, 310 SNP effects have variance of $10^{-3}\sigma_g^2$, and 2680 SNP effects have variance of $10^{-4}\sigma_g^2$, where $\sigma_g^2 = 100$. The other data set has more realistic SNPs effects based on the SNP effect distribution of Crohn's disease indicated in Wu et al. [2013]. We simulated the data set with dichotomous outcome with the proportion of cases to controls of 0.4. The minor allele frequencies were randomly sampled from the abdominal aortic aneurysms (AAA) data set, where AAA data set is a real genotype data of SNPs measured on 3104 individuals. There are 326,706 minor allele frequencies (MAF), which are ≥ 0.1, that can be sampled from.

Unrealistic Data

For the unrealistic data set, we ran the the Markov chain for 60,000 cycles with the first 10,000 samples discarded as warm-up. We drew every 50^{th} sample after warming-up. We first treated σ_g^2 as random. We tuned the variance of the proposal distribution (θ) manually. When $\theta = 1$, all the parameters mixed well. Then, we treated σ_g^2 as fixed. We set different values of σ_g^2: 0.1, 0.5, 1, 2, 5, 10, 30, 50, 100, and 500. Figure 3.4 shows the autocorrelation of the estimated parameters when $\sigma_g^2 = 10$: μ, three β's with the biggest absolute value, two randomly selected λ's, and two random selected Z's are all mixed very well. When σ_g^2 takes the other values, the autocorrelation plots show similar patterns. Figure 3.5 shows that when $\sigma_g^2 < 5$, the $\boldsymbol{\pi}$ parameters do not mix well and have high autocorrelation. But when $\sigma_g^2 \geq 5$, the $\boldsymbol{\pi}$ parameters mixed well.

We compared the BayesRB $\boldsymbol{\beta}$ estimates to the logistic regression $\boldsymbol{\beta}$ estimates, which are considered unbiased. Figure 3.6 shows that when σ_g^2 is set to a fixed value smaller than or equal to 10, the $\boldsymbol{\beta}$ estimates are under estimated, compared to the logistic regression

estimates. When σ_g^2 is set to a fixed value bigger than 10 ($\sigma_g^2 = 30, 50, 100, 500$) or is estimated using Metropolis-Hasting sampling, their $\boldsymbol{\beta}$ estimates of a given SNP are similar. In this situation, BayesRB shrunk the small SNP effects to zero, and estimated the bigger SNP effects similar to the logistic regression estimated $\boldsymbol{\beta}$.

From the recorded category that each SNP is assigned to, in each MCMC loop, we found that the relatively big effect SNPs are more often assigned to the category 2, 3 or 4. Figure 3.7 shows that when $\sigma_g^2 = 100$, more than 90% of the SNPs with the logistic regression estimated $\boldsymbol{\beta}$ bigger than 0.25 are assigned more than 95% of the time to category 2, 3 or 4 by BayesRB.

Realistic Data

For the realistic data set, we ran the the Markov chain for 20,000 cycles with the first 10,000 samples discarded as warm-up. We drew every 10^{th} sample after warming-up. We treated σ_g^2 as fixed. The autocorrelation plots show similar patterns as that using unrealistic data. When $\sigma_g^2 \geq 5$, all the parameters including $\boldsymbol{\pi}$ mixed well.

We also compared the BayesRB $\boldsymbol{\beta}$ estimates to the logistic regression $\boldsymbol{\beta}$ estimates. For the SNPs with the absolute value of the logistic regression $\boldsymbol{\beta}$ estimates smaller than 0.1, their BayesRB $\boldsymbol{\beta}$ estimates are shrunk to a value close to 0 (Figure 3.8a). For the SNPs with the absolute value of the logistic regression $\boldsymbol{\beta}$ estimates bigger than 0.1, the small σ_g^2 values ($\sigma_g^2 \leq 10$) will let the $\boldsymbol{\beta}$ underestimated, while big σ_g^2 values ($\sigma_g^2 = 30, 50, 100$, or 500) have $\boldsymbol{\beta}$ estimates similar to the logistic regression $\boldsymbol{\beta}$ estimates. (Figure 3.8c, d).

When the σ_g^2 is large enough ($\sigma_g^2 > 10$), different values of σ_g^2 affect the shrinkage level of the $\boldsymbol{\beta}$ estimates. Figure 3.9a shows that, for the SNPs with the logistic regression $\boldsymbol{\beta}$ estimates smaller than 0.1, the bigger the σ_g^2, the smaller the summation of the BayesRB $\boldsymbol{\beta}$ estimates squared. In other words, for the SNPs with the logistic regression $\boldsymbol{\beta}$ estimates smaller than 0.1, when $\sigma_g^2 > 10$, the bigger the σ_g^2, the more the BayesRB $\boldsymbol{\beta}$ estimates are shrunk. Figures 3.9b shows that, for the SNPs with the logistic regression $\boldsymbol{\beta}$ estimates smaller than 0.1, when $\sigma_g^2 = 100$, the overall distance of BayesRB $\boldsymbol{\beta}$ estimates and logistic $\boldsymbol{\beta}$ estimates is the smallest. In other words, when $\sigma_g^2 = 100$, the BayesRB $\boldsymbol{\beta}$ estimates for the big effect SNPs are closest to the logistic $\boldsymbol{\beta}$ estimates. Therefore, we set the $\sigma_g^2 = 100$ for the BayesRB analysis in genome-wide simulated data set and the real data set.

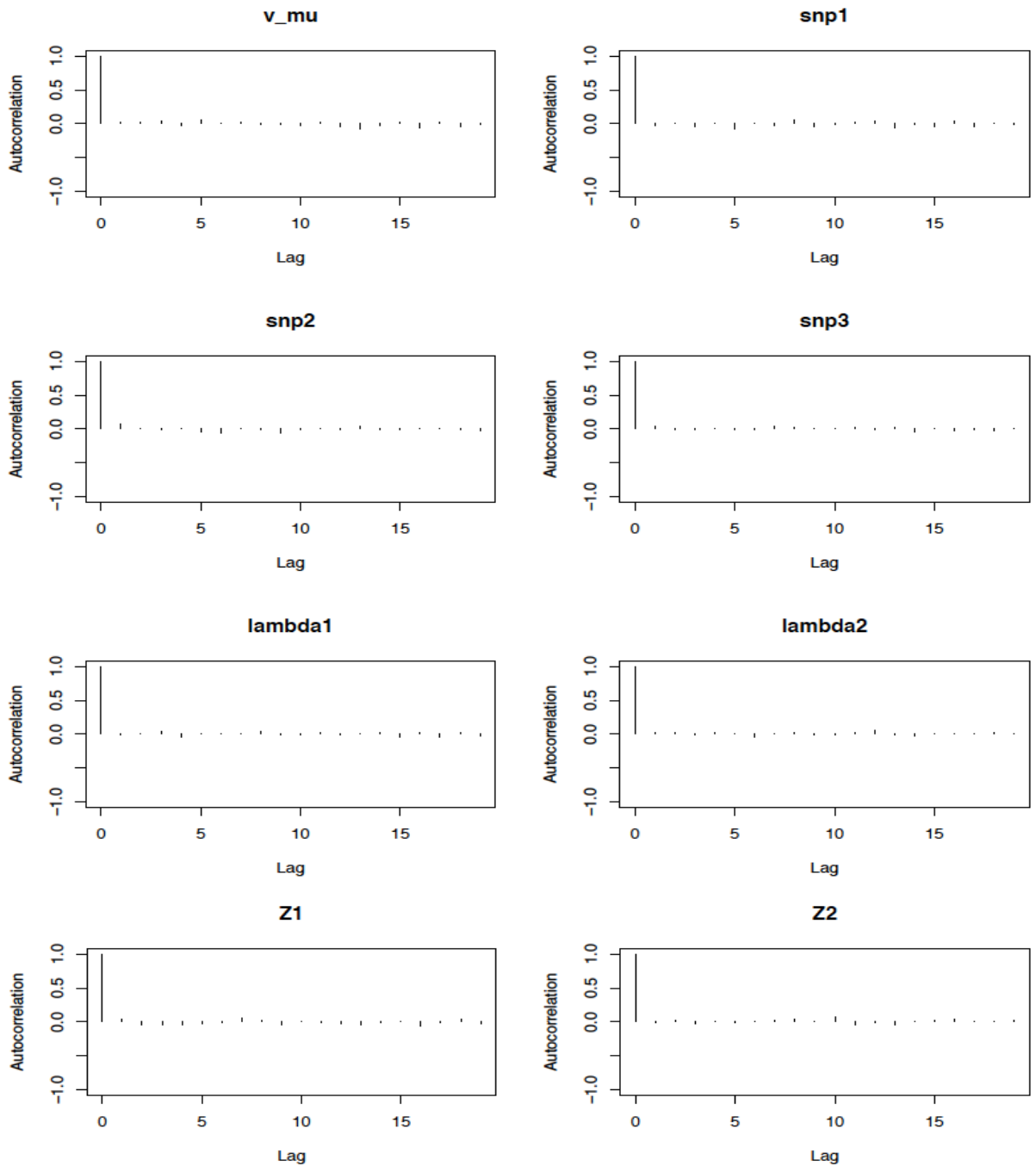


Figure 3.4: **Autocorrelation of the following parameters: μ, three β's with the biggest absolute values, two randomly selected λ's, two randomly selected Z's when $\sigma_g^2 = 10$.**

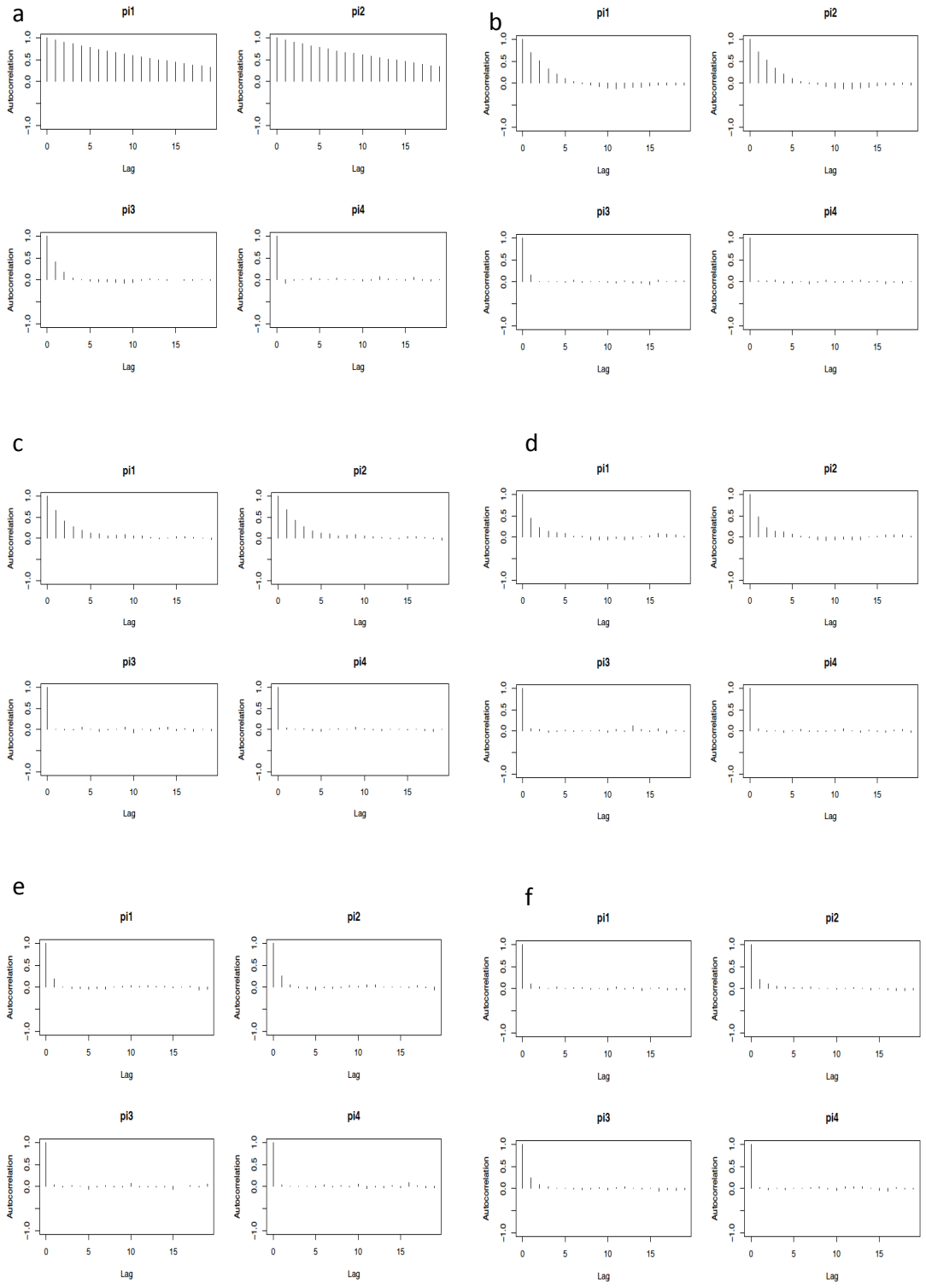
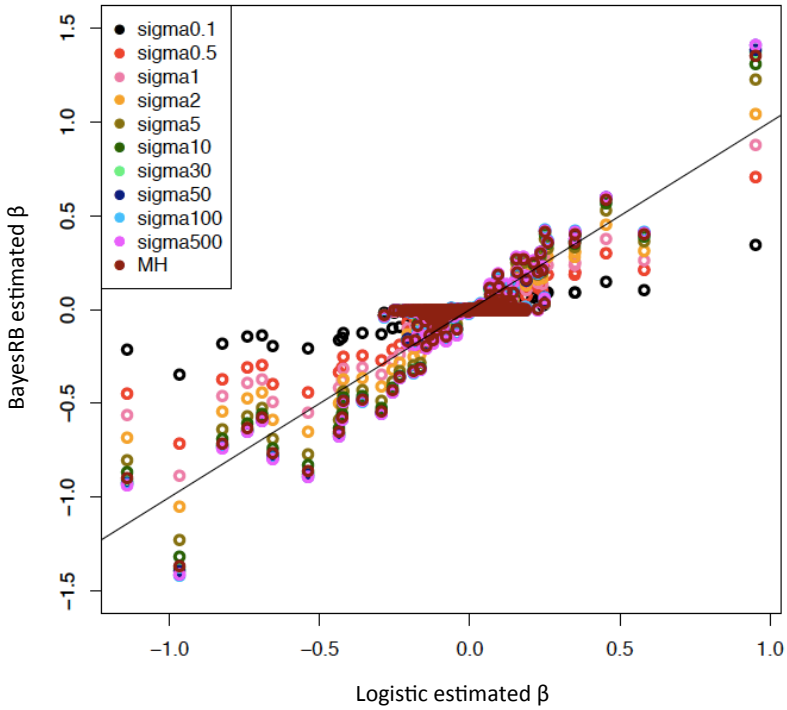


Figure 3.5: **Autocorrelation of π when σ_g^2 is 0.1 (a), 0.2 (b), 0.5 (c), 1 (d), 5 (e), and 10 (f).**

a.



b.

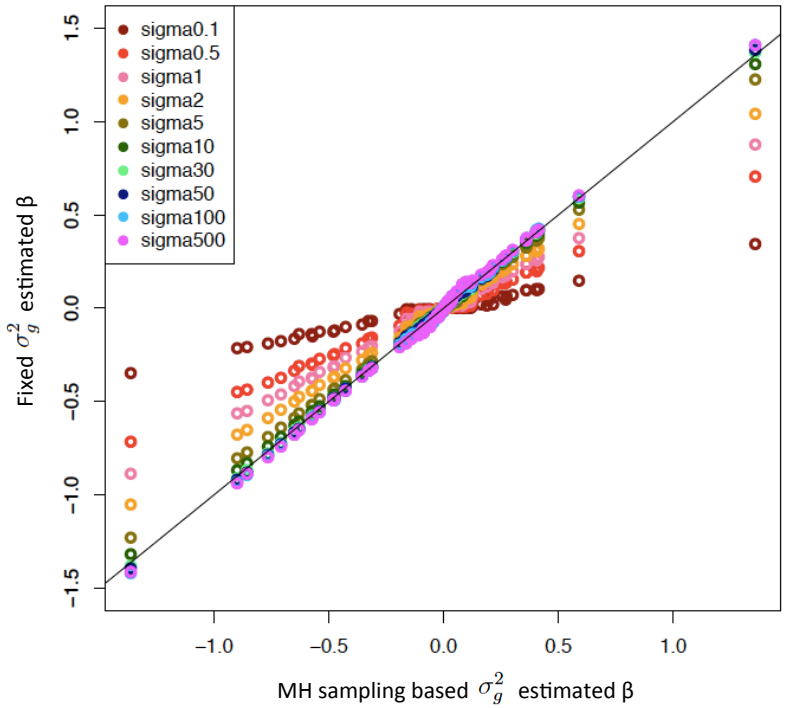


Figure 3.6: **A comparison of the SNP effect estimates under different σ_g^2 using the unrealistic data set.**

a. A comparison of the BayesRB β estimates to the logistic regression β estimates. **b.** A comparison of the β estimates with σ_g^2 updated by Metropolis-Hasting sampling to the β estimates with σ_g^2 fixed at different values. The solid lines in both plots are the diagonal lines.

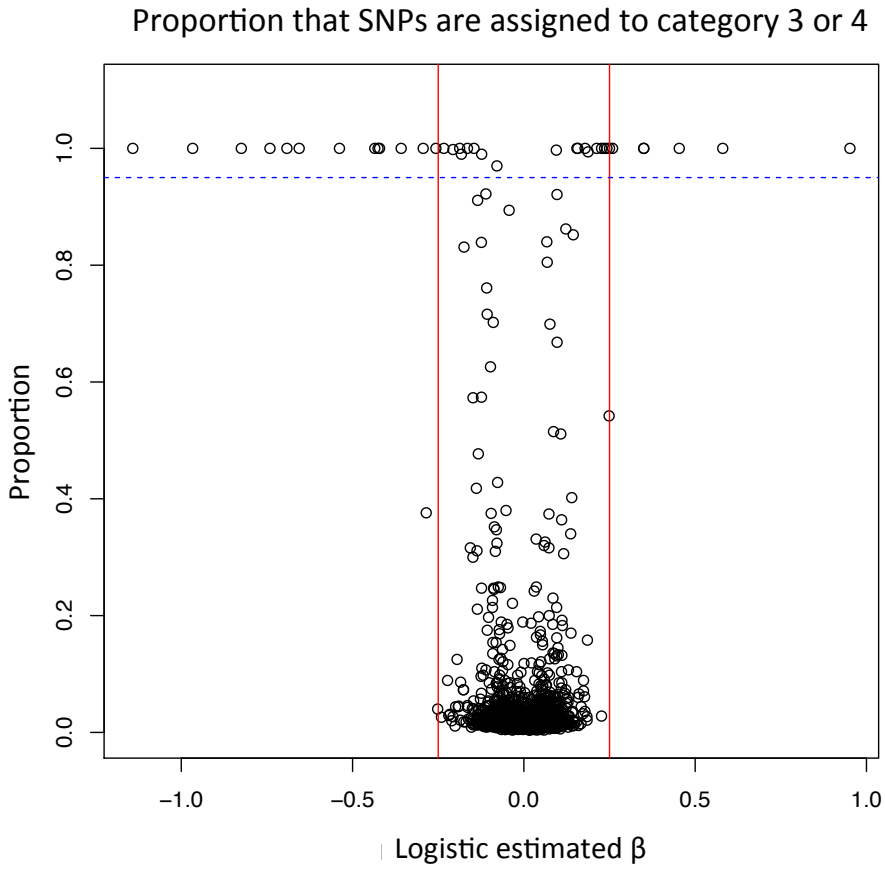


Figure 3.7: **The proportion of iterations that each SNP is assigned to category 2, 3 or 4, when $\sigma_g^2 = 100$.**

The blue dotted horizontal line indicates the proportion of 95% . The red vertical lines are drawn at -0.25 and 0.25.

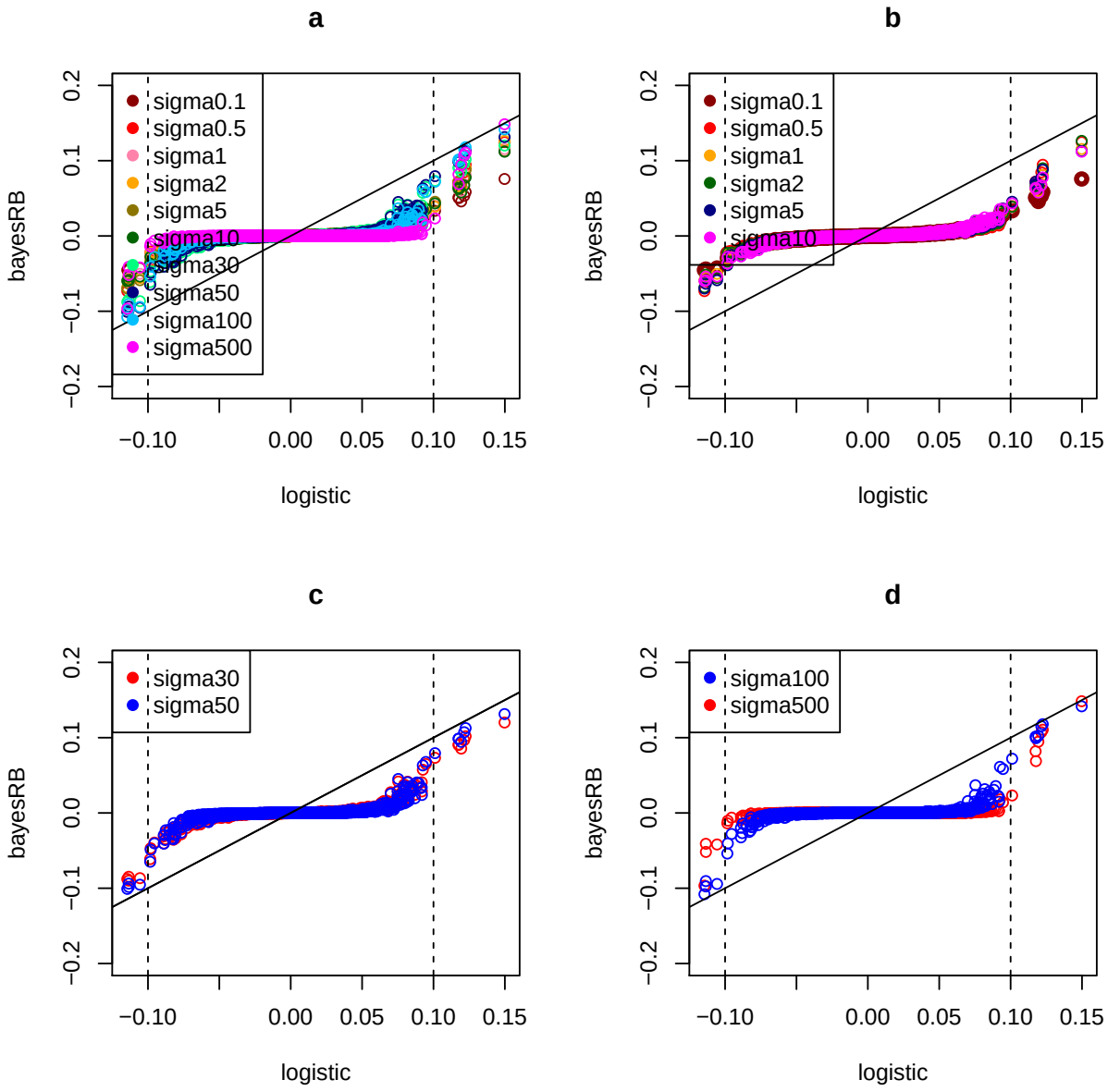
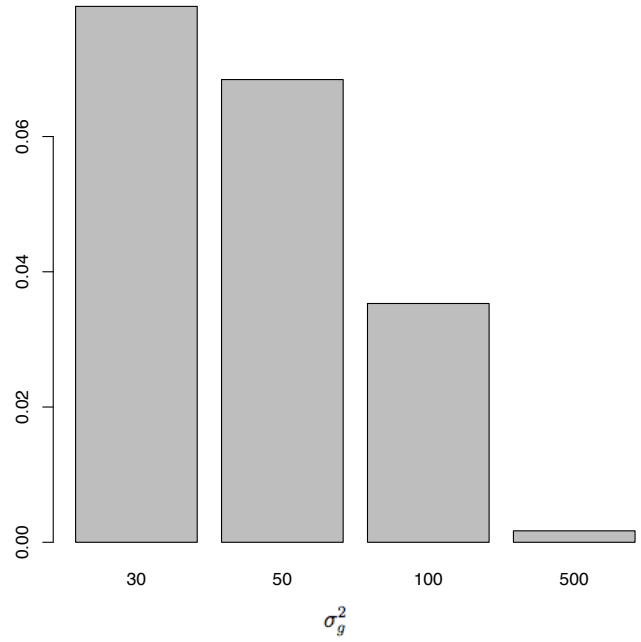


Figure 3.8: **A comparison of the SNP effect estimates under different σ_g^2 using the realistic data set.**

A comparison of the BayesRB β estimates to the logistic regression β estimates. The x axis label "logistic" indicates the SNP effects estimated by logistic regression. The y axis label "BayesRB" indicates the SNP effects estimated by BayesRB. The solid line in each plot is the diagonal line showing the equivalent values of logistic regression estimated β and the BayesRB estimated β. The dotted lines indicate the logistic regression estimated β of 0.1 and -0.1, respectively.

a. $\sum_{j=1}^{P}(\hat{\beta_{BRB}})^2$



b. $\sum_{j=1}^{P}(\hat{\beta_{BRB}} - \hat{\beta_{LR}})^2$

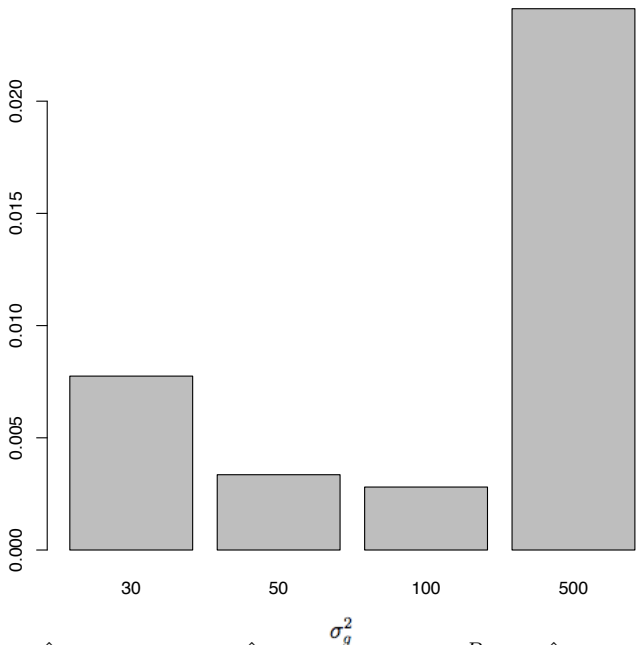


Figure 3.9: **a.** $\sum_{j=1}^{P}(\hat{\beta_{BRB}})^2$**, when** $\hat{\beta_{LR}} \leq 0.1$**. b.** $\sum_{j=1}^{P}(\hat{\beta_{BRB}} - \hat{\beta_{LR}})^2$**, when** $\hat{\beta_{LR}} > 0.1$.

In the formula, $\hat{\beta_{BRB}}$ is the BayesRB estimated SNP effect; and $\hat{\beta_{LR}}$ is the logistic regression estimated SNP effect.

3.3.3 Genome-wide Simulated data sets

To measure how well BayesRB works on genome-wide data sets with SNPs that are not fully independent, we applied BayesRB to genome-wide simulated data sets. We first simulated a population genetic data set containing 16,000 individuals' genotype data set using the software GWAsimulator [Li and Li, 2008] based on the HapMap phased CEU data. The genotype data contains all the 305,054 SNPs in the Illumina HumanHap300 SNP chip. We filtered out the SNPs in the data set with MAF smaller than 0.1. We assigned 195 causal SNPs that are not in the same LD blocks in the genome with effects follow a distribution described in Wu et al. [Wu et al., 2013]. The top 6 biggest effect causal SNPs (absolute values of the assumed true SNP effects ≥ 0.1) are defined as big effect causal SNPs. The 7 to 21 biggest effect causal SNPs ($0.05 \leq$ absolute values of the assumed true SNP effects < 0.1) are defined as the medium+ effect causal SNPs. The 22 to 34 biggest effect causal SNPs ($0.038 \leq$ absolute values of the assumed true SNP effects < 0.05) are defined as the medium- effect causal SNPs. Medium+ effect causal SNPs combined with medium- effect causal SNPs are medium effect causal SNPs. The rest SNPs (absolute values of the assumed true SNP effects < 0.038) are small effect causal SNPs. We obtained 1,500 cases and 1,500 controls from all the 16,000 individuals using the software GCTA [Yang et al., 2011], setting the heritability equal to 0.5 and disease prevalence equal to 0.1. We split each data set to 80/20 training and testing data sets, each with the same proportions of cases and controls.

First, we explored how close the genome-wide simulated data sets are to the real genome-wide data sets. Figure 3.10 is based on the logistic regression result of one of the genome-wide simulated data sets. In Figure 3.10a, the SNP effect density is close to real studies shown in Wu et al. [2013]. Figure 3.10b and c show that the big effect causal SNPs have smallest p values and the biggest SNP effect estimates. And there are peak towers around the big effect SNPs due to the LD, which is similar to a real GWAS manhattan plot. Therefore, the genome-wide simulated data sets are similar to the real genome-wide data sets.

For each BayesRB analysis on genome-wide simulated data sets, we ran the Markov chain for 15,000 cycles with the first 5,000 cycles as warm-up. We drew every 20 sample after warming-up. We made diagnostic plots. The autocorrelation plots of most parameters show

similar patterns as Figure 3.4, which indicates that the parameters are mixed well. But some do not. A larger number of MCMC loops and the bigger thinned number is required. But considering the computational time, we still used those estimates in the following analysis, although they are not ideal to use.

We randomly selected one data set out of 50 and compared the SNP effect estimation performance of BayesRB to BayesR, logistic regression and LASSO method using this data set. Since LASSO underestimates the selected associated SNP effects, we reestimated the SNP effects by using the marginal logistic regression analysis. Then, LASSO and logistic regression methods provide the same SNP effect estimates. So, we only showed logistic regression result here. Figure 3.11a shows that BayesRB estimates have a linear relationship with the BayesR estimates, but BayesR has shrunk estimates. Figure 3.11b shows that the SNPs with big logistic regression estimates have BayesRB estimates close to logistic regression estimates. SNPs with small logistic regression estimates have BayesRB estimates close to 0. The red dots, which indicate the big effect and the medium+ effect causal SNPs, show that the top 21 biggest effect causal SNPs have relatively big effect estimates. Figure 3.12 shows the proportion of the loops that the SNPs are assigned to the category 2, 3 or 4. All the big effect and the medium+ effect causal SNPs have relatively big proportions, which are bigger than 0.2.

We also compared the performance of identifying associated SNPs of BayesR, BayesRB, logistic regression and LASSO. Comparison between methods are assessed on their ability to identity genomic region of 50 SNPs window containing causal SNPs. The reason we used 50 SNPs window is because we did not have the SNP manifest identifying the SNP locations for the Illumina HumanHap300 phased data set we used in the GWASimulator software. And on the Illumina website, they indicated "Although assays on the Human Hap300 BeadChip were chosen using tagSNPs, SNPs are evenly spaced across the genome to ensure comprehensive coverage. On average, there is 1 SNP every 9 kb across the genome (median spacing = 5kb). The average 90th percentile gap on the HumanHap300 BeadChip is 19kb." Therefore, it is reasonable to set the window size as that containing 50 SNPs. We calculated the true positive rate (TPR) and the false positive rate (FPR) of the four methods to detect the windows containing causal SNPs in each replicate. The TPR is the proportion of windows

containing causal SNPs that are correctly identified as containing associated SNPs. The FPR is the proportion of windows not containing causal SNPs but are incorrectly identified as containing associated SNPs. Figure 3.13 shows that all the methods have mean TPR of 1 to detect the windows containing the five big effect causal SNPs. (One big effect causal SNP is excluded in the QC process, thus is not shown in the result.) When the FPR < 0.015, BayesR has the biggest mean TPR to detect the windows containing medium effect causal SNPs. When the FPR $>= 0.015$, LASSO has the biggest mean TPR. BayesRB has the lower, but not much lower, mean TPR compared to the other methods under the same FPR.

Then, we compared the power of BayesR, BayesRB, logistic regression and LASSO of having each window containing causal SNPs identified within the 50 replicates. The power is calculated using the number of replicates in which a window containing a given SNP is identified divided by the total number of replicates where at least one SNP in the window have genotype data. Figure 3.14a and Figure 3.15a, b, c, and d show that using the thresholds under the same FPR of 0.001, the power of the four methods are similar. LASSO has slightly bigger power for the medium+ effect causal SNPs than the others. BayesRB has slightly bigger power for the medium- effect causal SNPs and the small effect causal SNPs than the others. Figure 3.14b and Figure 3.15e, f, g, and h show that using the thresholds under the same FPR of 0.05, LASSO has the biggest power, followed by BayesRB, while BayesR has the lowest power for all the causal SNPs. The power of all the methods detecting the big effect SNPs are 1.

At last, we compared the performance of risk prediction of BayesR, BayesRB, logistic regression and LASSO on the testing data sets by comparing the area under the curves (AUCs). Figure 3.16 shows that BayesRB and BayesR generate bigger AUCs than LASSO and logistic regression. But BayesR's median value of AUC is slightly higher than BayesRB and BayesR's prediction is more precise than BayesRB.

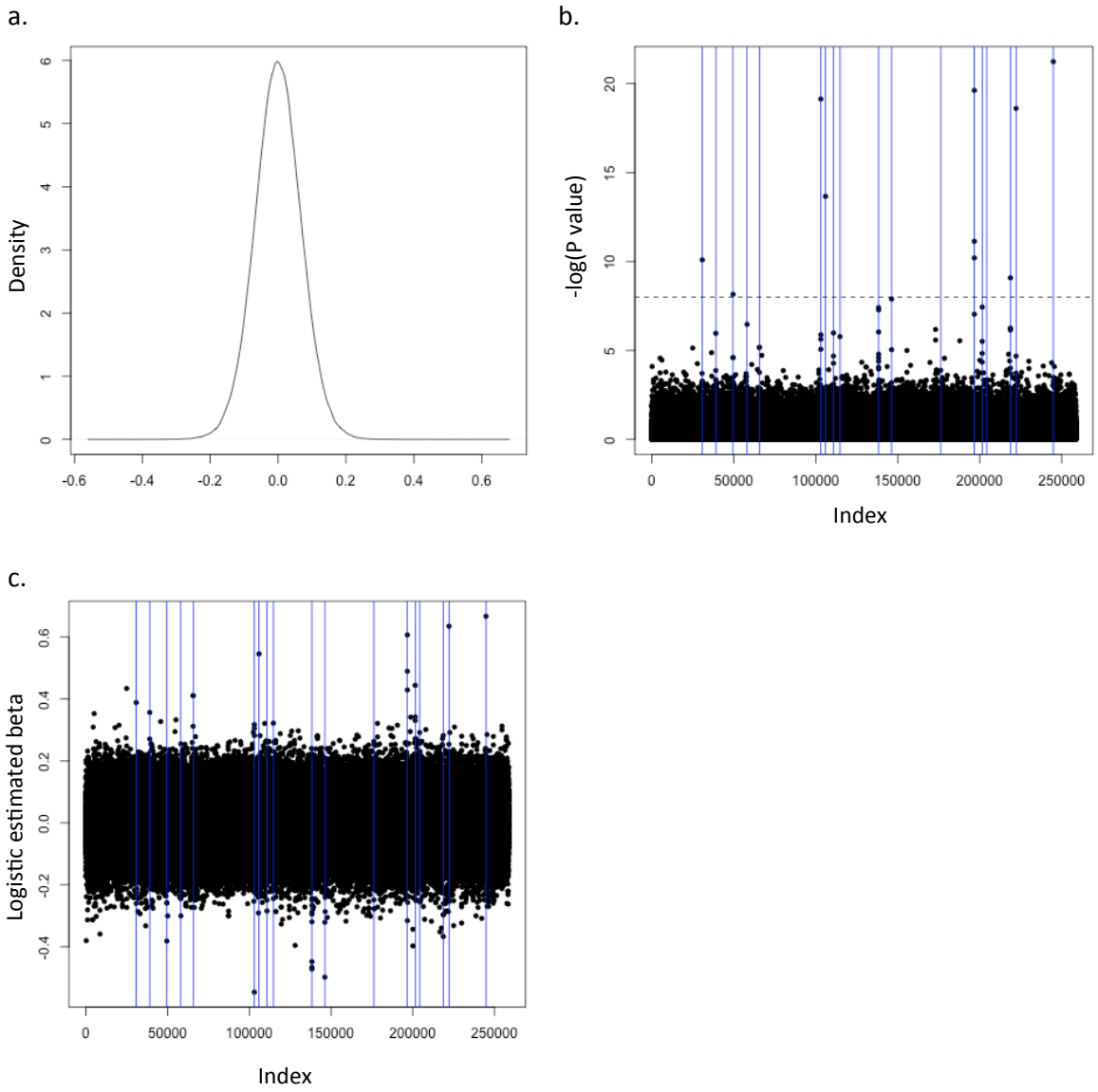


Figure 3.10: **Evaluation of the genome-wide simulated data sets by visualizing the distribution of SNP effect estimates and $-log_{10}$(P values) generated by logistic regression method.**

a. Density plot of the logistic regression SNP effect estimates. **b.** Scatter plot of $-log_{10}$(P values) generated by logistic regression method. The vertical blue lines indicate the top 21 SNPs with the biggest causal SNP effects (big effect causal SNPs and the medium+ effect causal SNPs). The horizontal dotted line indicates $-log_{10}(P) = 5 \times 10^8$. **c.** Scatter plot of SNP effect estimates by logistic regression method. The vertical blue lines have the same meaning as b.

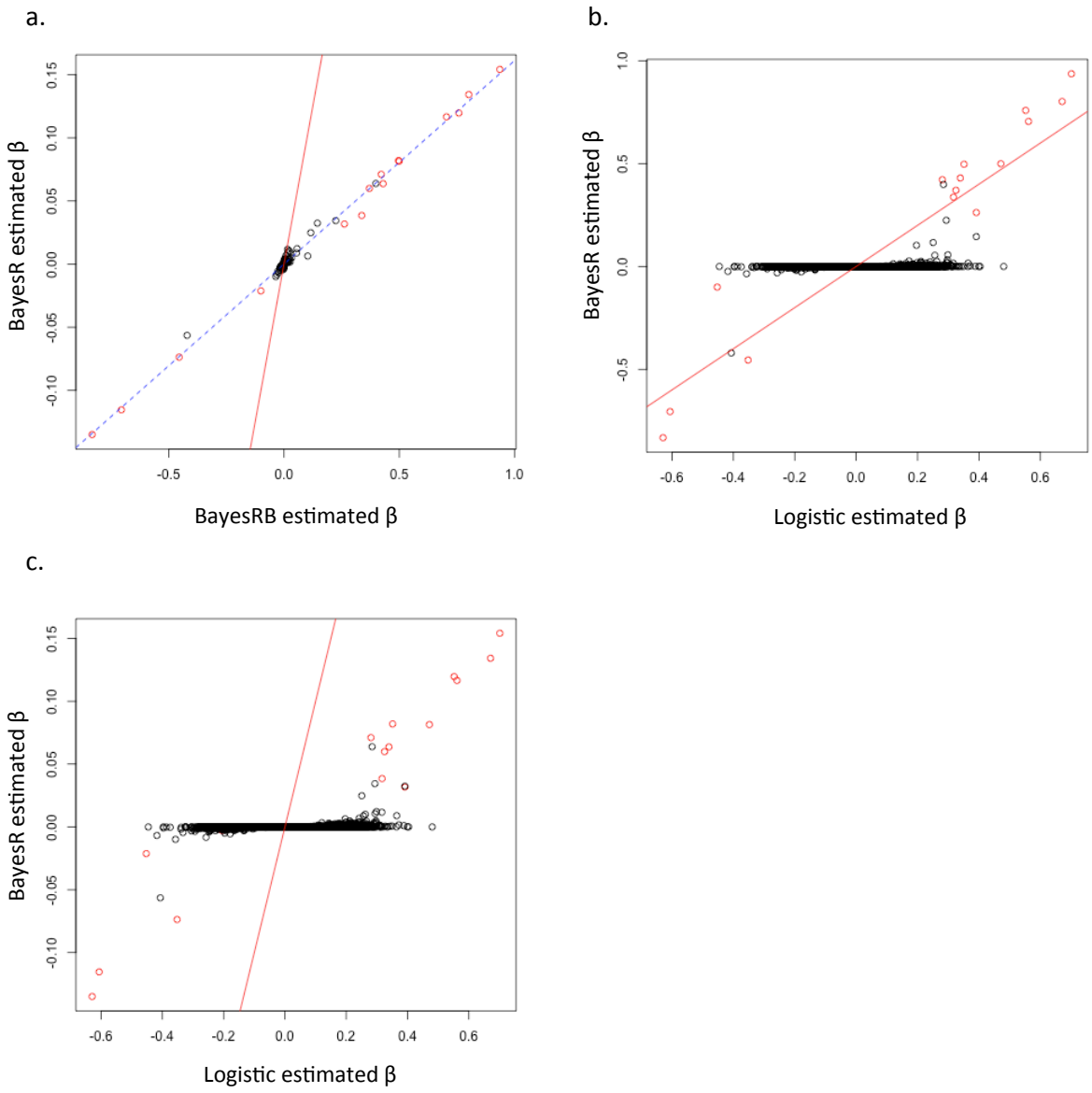


Figure 3.11: **The comparisons of SNP effect estimates by BayesRB, BayesR, and logistic regression detected in one of the genome-wide simulated training data sets.**

The red dots indicate the SNP effect estimates of the 21 biggest effect SNPs (big effect causal SNPs and the medium+ effect causal SNPs). The red solid line is the diagonal line. **a.** BayesRB SNP effect estimates vs. BayesR SNP effect estimates. The blue dotted line is the fitted regression line. **b.** Logistic regression SNP effect estimates vs. BayesRB SNP effect estimates. **c.** Logistic regression SNP effect estimates vs. BayesR SNP effect estimates.

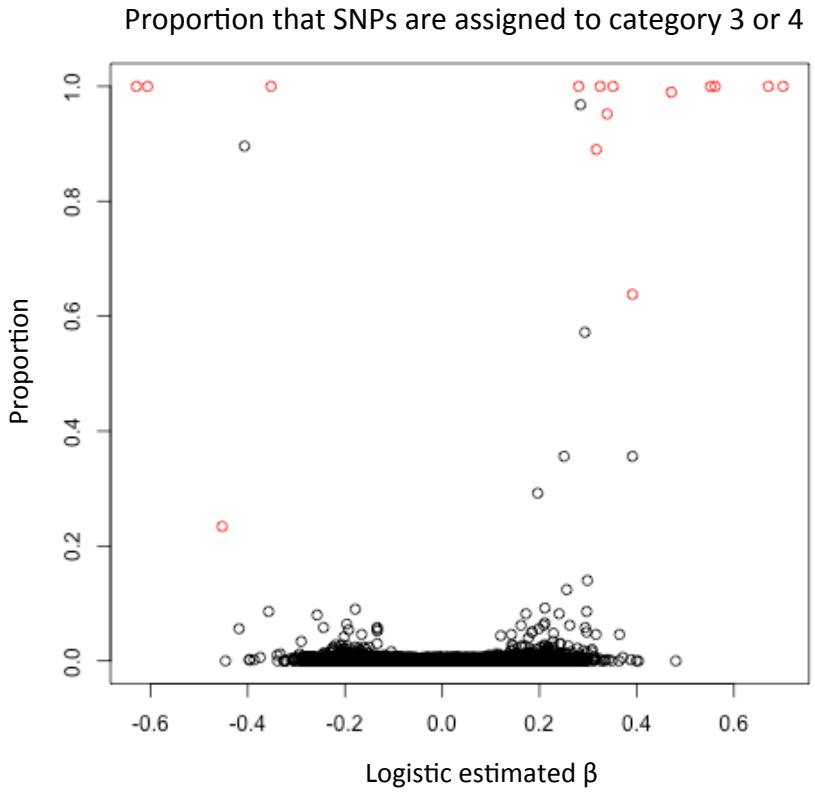


Figure 3.12: **The proportion of the iterations that the SNPs are included in the model in one of the genome-wide simulated training data sets.**

The red dots indicate the 21 biggest effect SNPs (big effect causal SNPs and the medium+ effect causal SNPs).

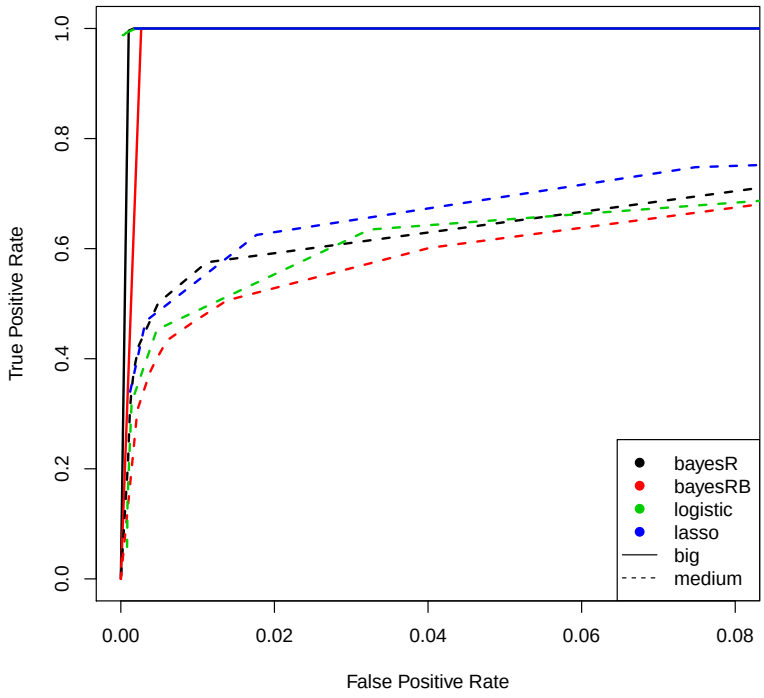


Figure 3.13: **True positive rate vs. false positive rate to detect the windows containing the big effect causal SNPs and the medium effect causal SNPs in the genome-wide simulated data sets.**

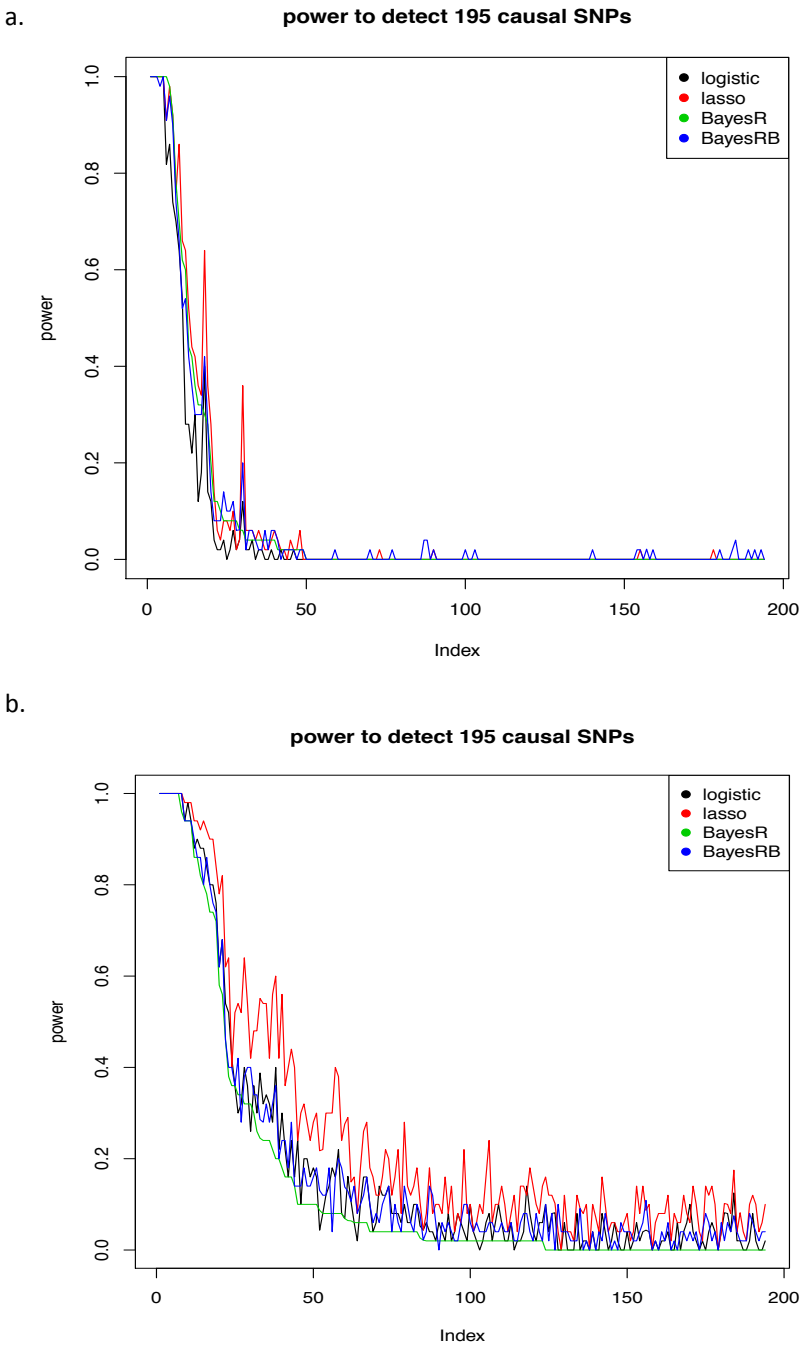


Figure 3.14: **Power of the windows containing the causal SNPs being detected within 50 replicates in the genome-wide simulated data sets under the FPR of 0.001 (a) and 0.05 (b). The x axis is the SNP index ordered by the causal SNP effect sizes.**

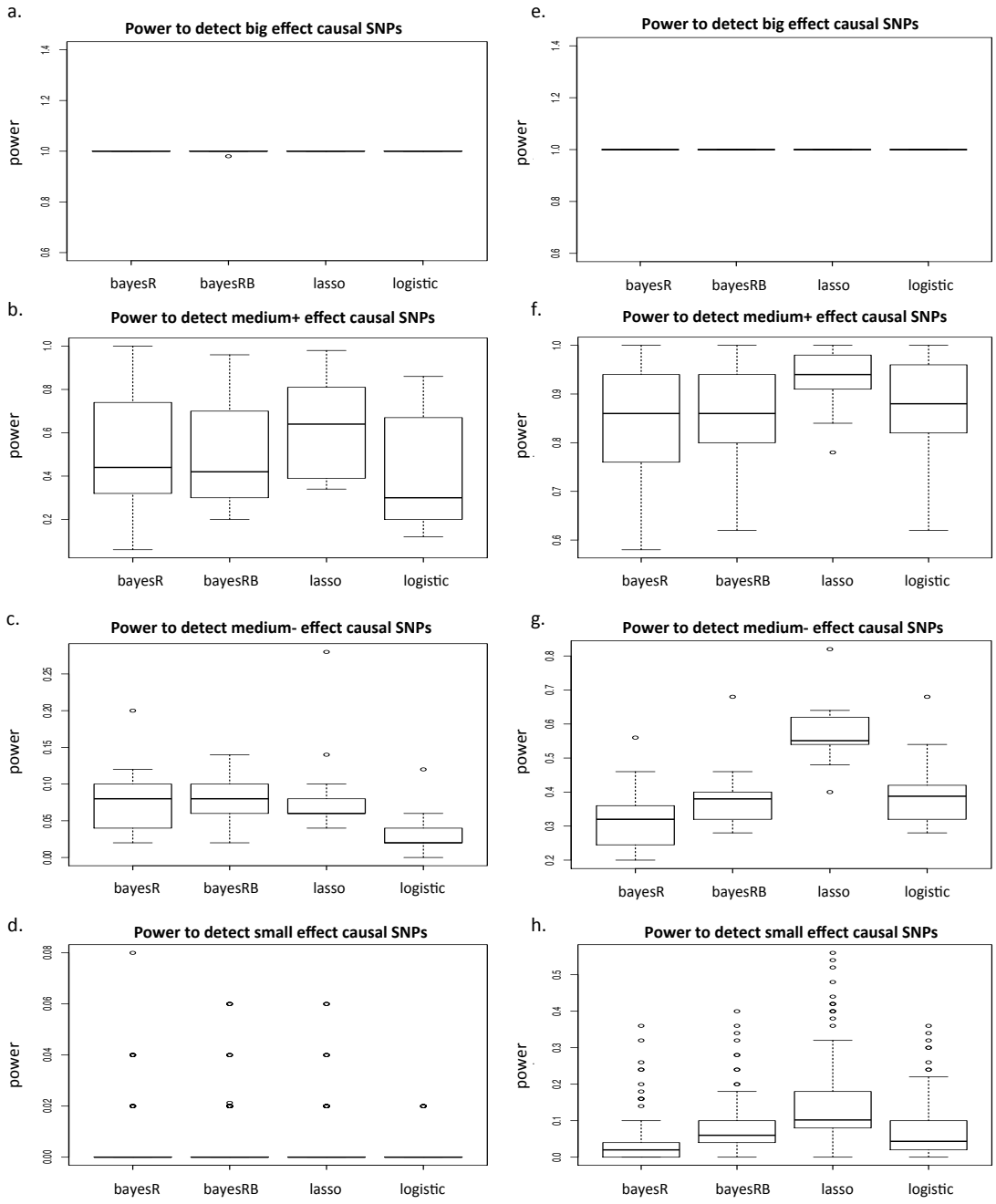


Figure 3.15: **Power of the windows containing big effect causal SNPs (a, e), medium+ effect causal SNPs (b, f), medium- effect causal SNPs (c, g), and small effect causal SNPs (d, h) being detected within 50 replicates under the FPR of 0.001 (a, b, c, d) and 0.05 (e, f, g, h) in the genome-wide simulated data sets.**

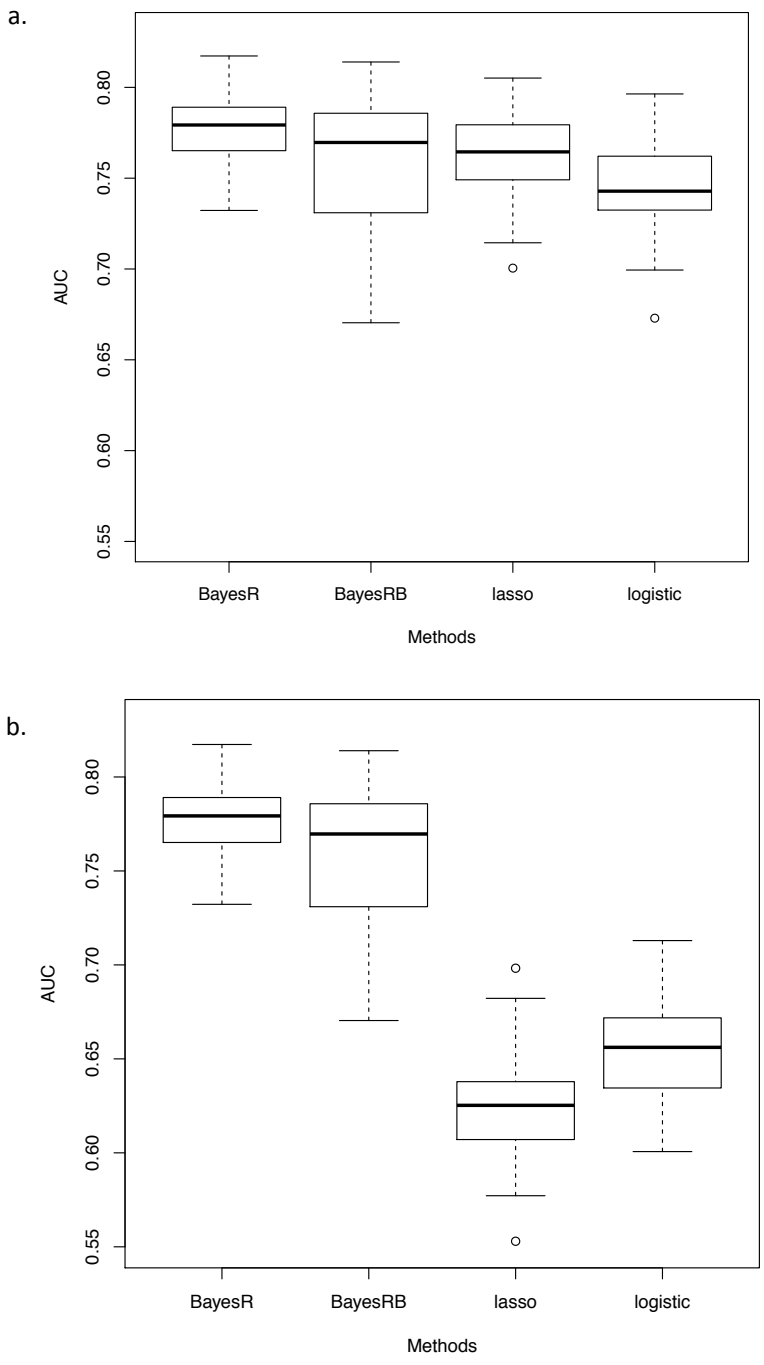


Figure 3.16: **Area under the curve (AUC) of 50 replicates in the genome-wide simulated data sets. Logistic regression and LASSO use the thresholds when FPR = 0.001 (a) and FPR = 0.05 (b).**

3.3.4 WTCCC data sets

We assessed the performance of BayesRB for seven diseases of the WTCCC data set. First, we converted the data sets to PLINK format using MEGA2 [Mukhopadhyay et al., 2005]. Then, following the analyses of Moser et al., we performed strict quality control (QC) on SNP data using PLINK [Purcell et al., 2007]. We removed individuals with missing genotypes greater than 2%. We removed loci with the minor allele frequencies smaller than 1% and SNPs with missingness bigger than 1% for each of the 7 case data sets and the two control data sets. We combined each case and the two control sets into 7 trait case-control studies. We removed the SNPs significant at 5% for differential missingness between cases and controls and SNPs significant at 5% for Hardy-Weinberg equilibrium. We also took relatedness testing using a pruned set of SNPs with LD of r^2 smaller than 0.05. We used the software PRIMUS [Staples et al., 2014] to identify a maximum unrelated set of individuals. We conducted principle component analysis (PCA) using the software GCTA [Yang et al., 2011] and removed individuals in each disease who had poor clustering by visual inspection. Then, we used the software BEAGLE [Browning and Browning, 2016] to fill in the missing genotypes. After QC, the data included 1665 cases of bipolar disorder (BD), 1882 cases of coronary artery disease (CAD), 1576 cases of Crohn's disease (CD), 1805 cases of hypertension (HT), 1721 cases of rheumatoid arthritis (RA), 1850 cases of type 1 diabetes (T1D), 1761 cases of type 2 diabetes (T2D), and 2757 to 2782 controls depending on the traits. The number of genotypes ranged from 306,702 for BD to 312,035 for CD.

Moser et al. [2015] applied BayesR to all the above seven case-control data sets, treating the binary outcome as the response in an ordinary linear regression. We only used the CD and BD data sets. We chose CD and BD data sets for the following two reasons: 1) both diseases have several significant SNPs contributing to the diseases together, unlike RA or T1D, who are largely influenced by MHC region [Burton et al., 2007]; 2) in Moser et al., BayesR has relatively better performance on these two data sets than the others, except RA and T1D data sets. For both CD and BD data sets, we split each one to 80/20 training and testing data sets for 20 replicates with the same proportions of cases and controls.

Before applying BayesRB to the data sets, we diagnosed the two cleaned data sets to double check the quality control (QC) results. We plotted the Manhattan plots (Figure

3.17a,c) and the QQ plots (Figure 3.17b,d) for the two data sets, and compared the plots as well as the significant SNPs to the previous studies. The Manhattan plots and QQ plots, as well as the significant SNPs match the results in previous studies [Burton et al., 2007; Bowden and Dudbridge, 2009]. Therefore, the two data sets are good to use.

Both CD and BD data sets have around 45% more individuals and 20% more SNPs than the genome-wide simulated data sets have. Therefore, considering the computational time, we ran the Markov chain for 13,000 cycles with the first 5,000 cycles as warm-up. We drew every 20 sample after warming-up. Then, we made diagnostic plots. Most of the parameters show similar patterns as Figure 3.4, which indicate that those parameters are mixed well. But some SNP effects do not mix so well, which require larger number of MCMC loops and the bigger thinned number. In the following analysis, we still used those estimates, although they are not ideal to use.

Crohn's Disease (CD) data set

First, we randomly selected one training data set out of 20 to compare the SNP effect estimation performance and the associated SNP detection performance of BayesRB, BayesR and logistic regression. Figure 3.18a shows that the BayesRB SNP effect estimates have a linear relationship with the BayesR estimates, except for two SNPs. The two SNPs are rs7593114 and rs6715049, both located on chromosome 2. Figure 3.18b compares the marginal logistic regression estimates to both BayesRB and BayesR estimates of the top 10 SNPs with the biggest BayesRB effects. Except for the two SNPs, BayesRB has slightly smaller estimates than marginal logistic regression, while BayesRB has much smaller estimates than the marginal logistic regression. The two SNPs which do not locate around the diagonal line are rs11887827 and rs6715049 on chromosome 2. Then, we generated LocusZoom plots [Pruim et al., 2010] to show the relationship of rs11887827, rs6715049 and rs7593114. Figure 3.19a shows that the three SNPs are highly correlated ($r^2 > 0.9$), but while rs11887827 has a p value as small as 1.701×10^{-5}, rs6715049 and rs7593114 do not have significant p values. Then, we conducted the logistic regression conditioning on rs11887827. Figure 3.19b shows that with rs11887827 conditioned on, rs6715049 and rs7593114 are pumped up with p values

as small as 7.188×10^{-20} and 9.189×10^{-11}, respectively. We recompared the SNP effect estimates of BayesRB to logistic regression, but with rs11887827 conditioned on. Figure 3.18c shows that all the top 10 SNPs' BayesRB estimates are close to the logistic regression estimates, but slightly smaller than the logistic regression estimates. Then, we plotted manhattan plots based on the p values from both marginal logistic regression (Figure 3.20a, b) and logistic regression with rs11887827 conditioned on (Figure 3.20c, d). We explored the corresponding p values from logistic regression of the top 10 SNPs with the biggest BayesRB estimates (Figure 3.20a, c) and the biggest BayesR estimates (Figure 3.20b, d). Most of the top 10 SNPs locate on the top of the manhattan tower. Before rs11887827 is conditioned on, rs7593114 and rs6715049 locate at the bottom of the manhattan tower. After the rs11887827 is conditioned on, rs7593114 locates on the top of the manhattan tower, while rs6715049 is halfway up the tower. Table 3.2a shows that while rs11887827 and rs7593114 have perfect LD in the controls, 115 off diagonal entries in the cases. Table 3.2b shows that while only 5 out of 2225 controls having homozygous rs11887827 mutation and heterozygous rs6715049 mutation, 114 out of 1260 cases having this pattern. In Figure 3.20c, d, it seems that another SNP rs903228 is also halfway up the same tower of rs11887827. But actually, this SNP is 2,797kb away from rs11887827. We also investigated manhattan towers on chromosome 1 and 16. In Figure 3.20c, rs7515029 on chromosome 1 does not locate on the top of the manhattan tower. The SNP on the top of the tower is rs2201841, which is one of the top 30 biggest BayesRB estimated SNPs. rs7515029 and rs2201841 have low correlation (Figure 3.21a). In Figure 3.20c and d, two SNPs on the same tower in chromosome 16 have large BayesRB and BayesR estimates. But according to the Figure 3.21b, although the two SNPs are close to each other, they locate on different genes with a low correlation ($r^2 < 0.2$). Therefore, it is reasonable that BayesRB and BayesR generate big estimates for both of the SNPs.

We calculated the TPR and FPR to detect the 250kb windows containing the 201 previously reported SNPs. The 201 previously reported SNPs are obtained from GWAS Catalog [Welter et al., 2014] and Liu at al. [2015]. Only the SNPs reported by studies with European ancestry samples are included. All 201 SNPs have reported p values smaller than 5×10^{-8}. Figure 3.22a and b show that logistic regression has the biggest TPRs, followed by LASSO,

while BayesR has the lowest TPR, under the same value of FPRs. When the FPR is 0.08, the TPR is around 0.2 for all the methods.

Then, we compared the power of BayesR, BayesRB, logistic regression and LASSO to have each window containing causal SNPs identified within the 20 replicates. Figure 3.22c and d show that using the thresholds under the same FPR of both 0.001 and 0.05, logistic regression has highest power to detect the previously reported SNPs, and LASSO has the lowest power. For some SNPs, BayesRB has bigger power than BayesR, but for other SNPs, it does not.

At last, we compared the performance of risk prediction of BayesR, BayesRB, logistic regression and LASSO on the testing data set by comparing the AUCs. Figure 3.22e and f show that BayesRB and BayesR generate bigger AUCs than LASSO and logistic regression. But BayesR's median value of AUC is slightly higher than BayesRB. BayesRB's prediction is more precise than BayesR.

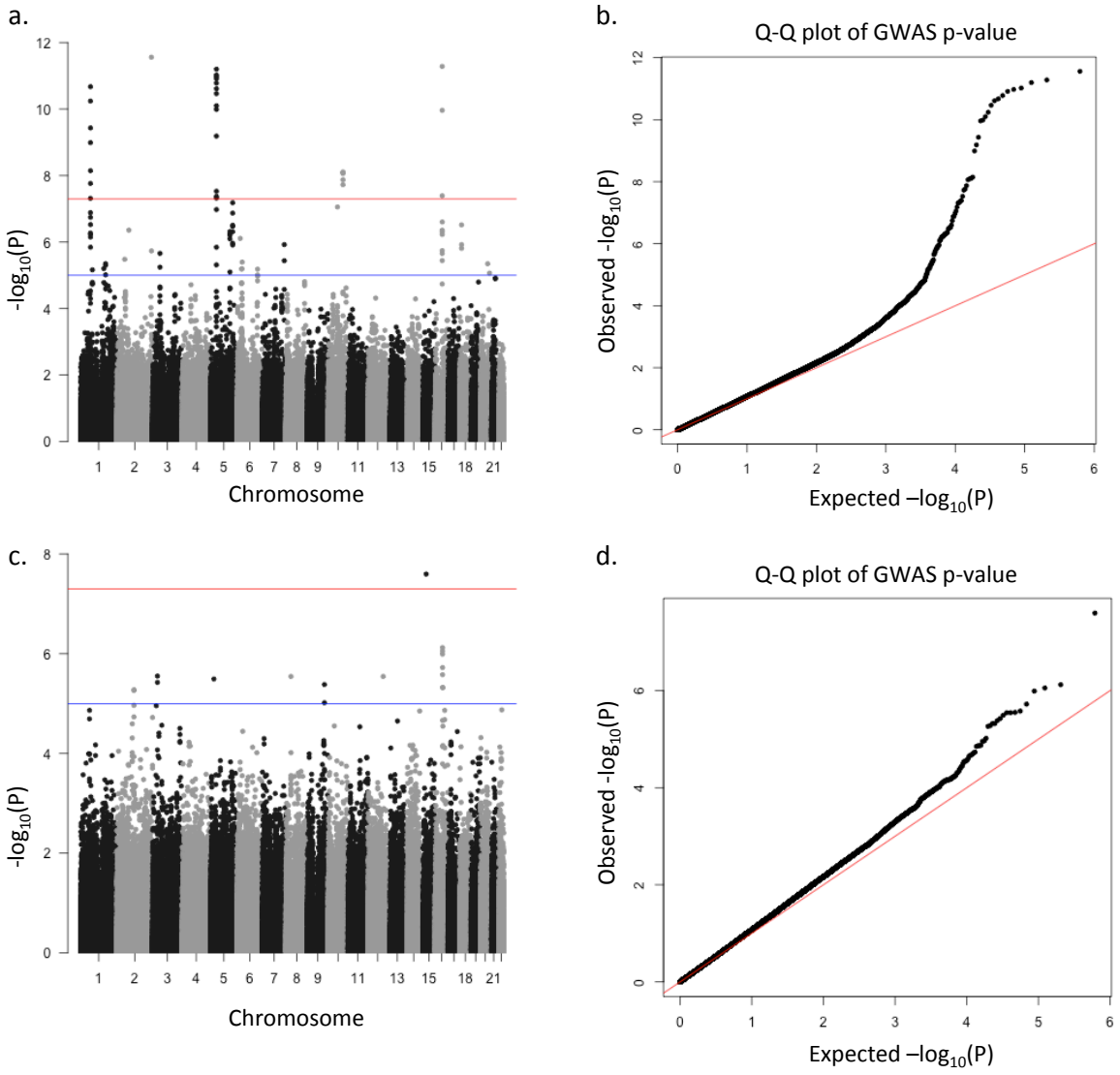


Figure 3.17: **Manhattan plots (a, c) and the QQ plots (b, d) for the CD data set (a, b) and the bipolar disorder data set (c, d).**

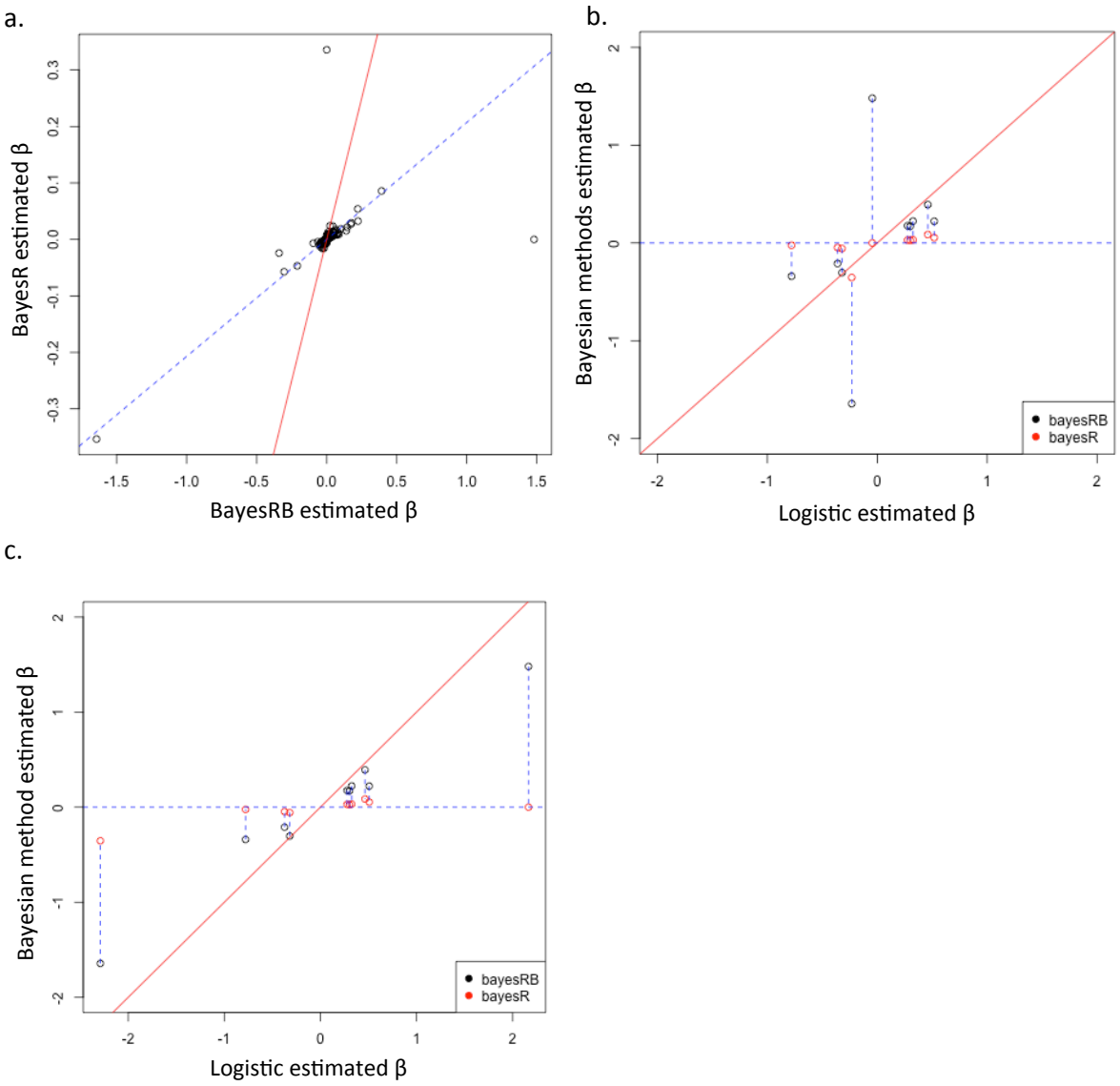


Figure 3.18: **The comparisons of SNP effect estimates by BayesRB, BayesR, and logistic regression in one of the CD training data sets.**

The red solid lines are the diagonal lines. **a.** Compare the SNP effect estimates of BayesRB to BayesR. The blue dotted line is the fitted regression line, excluding the estimates of rs6715049 and rs7593114. **b.** Compare the marginal logistic regression estimates to BayesRB and BayesR estimates of the top 10 SNPs which have the biggest BayesRB estimates. **c.** Compare the logistic regression estimates conditioning on rs11887827 to BayesRB and BayesR estimates of the top 10 SNPs which have the biggest BayesRB estimates.

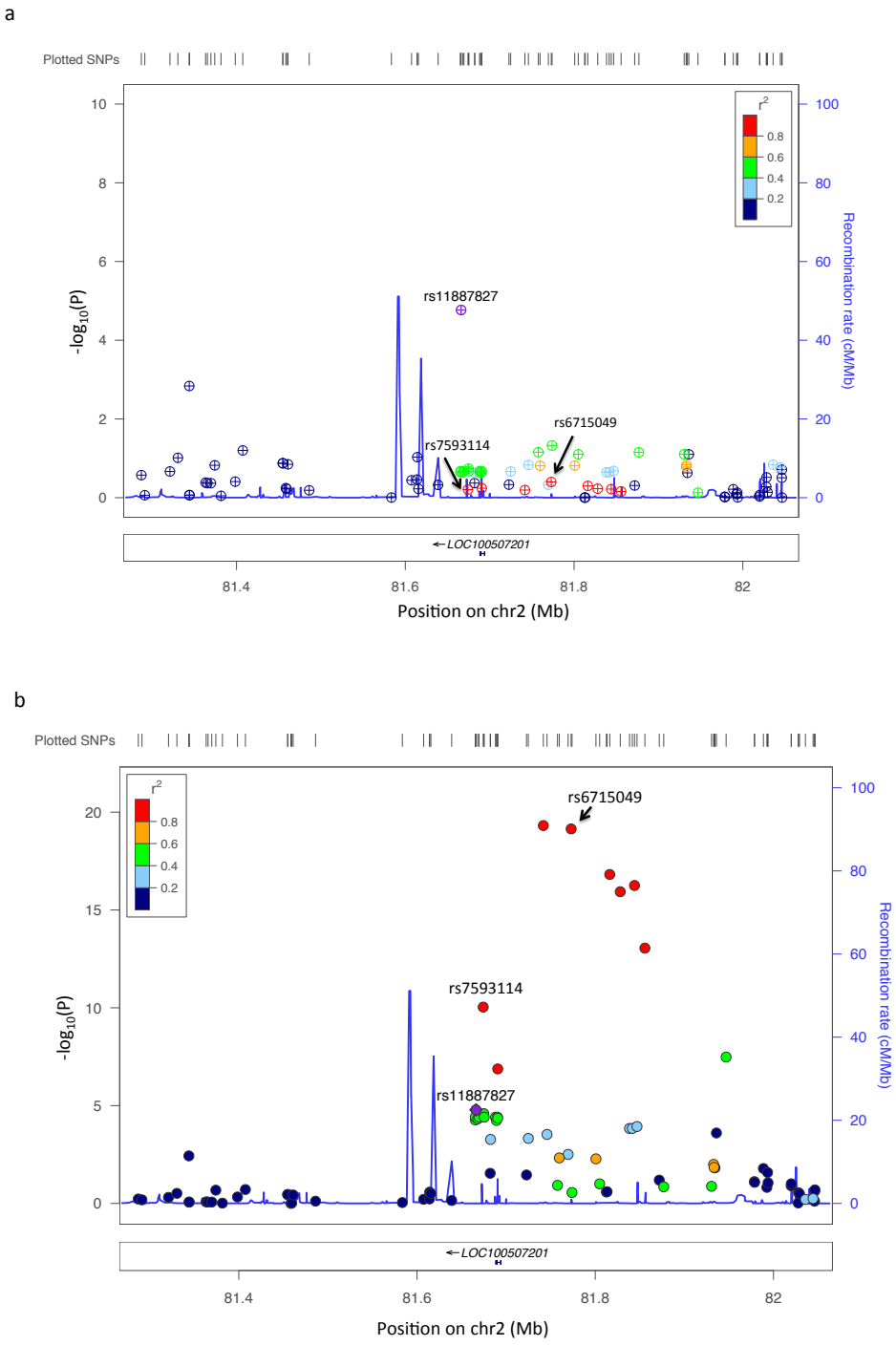


Figure 3.19: **LocusZoom plots of 400kb region on chromosome 2, centered on rs11887827.**

a. The p values are generated from marginal logistic regression analysis. **b.** The p values are generated from logistic regression analysis with rs11887827 conditioned on.

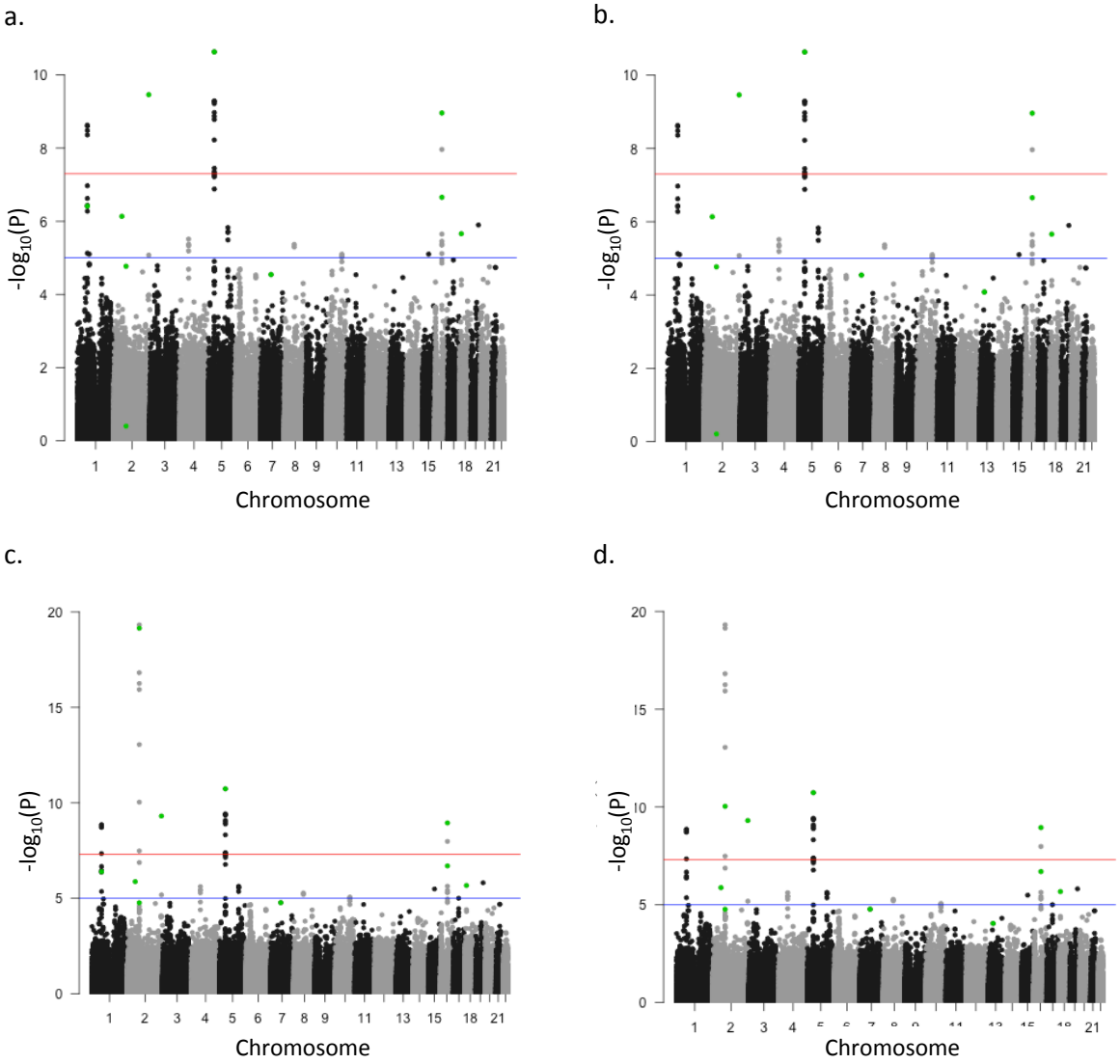


Figure 3.20: **Manhattan plots of one of the CD training data sets before and after the SNP rs11887827 is conditional on with the top 10 BayesRB and BayesR estimated SNPs highlighted, respectively.**

a. The y axis is the $-log_{10}$ of the marginal p values. The green dots indicate the 10 SNPs with the biggest BayesRB SNP effect estimates. **b.** Same manhattan plot as in **a**. The green dots indicate the 10 SNPs with the biggest BayesR SNP effect estimates. **c.** The y axis is the $-log_{10}$ of the p values conditioned on rs11887827. The green dots indicate the 10 SNPs with the biggest BayesRB SNP effect estimates. **d.** Same manhattan plot as in **c**. The green dots indicate the 10 SNPs with the biggest BayesR SNP effect estimates.

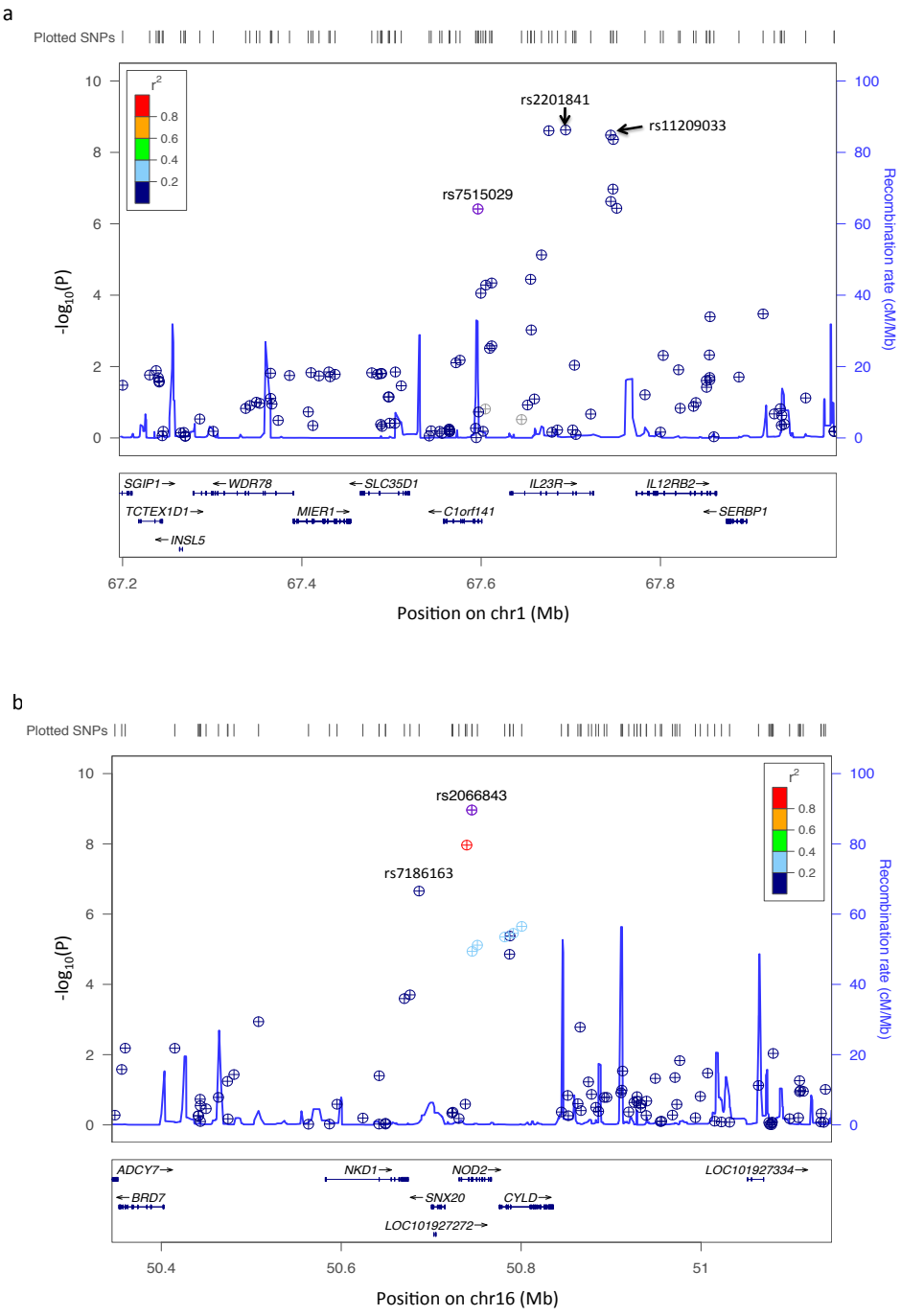


Figure 3.21: **LocusZoom plots of 400kb region, centered on rs7515029 and rs2066843, respectively.**

The purple dot indicate the SNP that the plot is centered on. The p values are generated by marginal logistic regression. **a.** LocusZoom plot of 400kb region on chromosome 1, centered on rs7515029. **b.** LocusZoom plot of 400kb region on chromosome 16, centered on rs2066843.

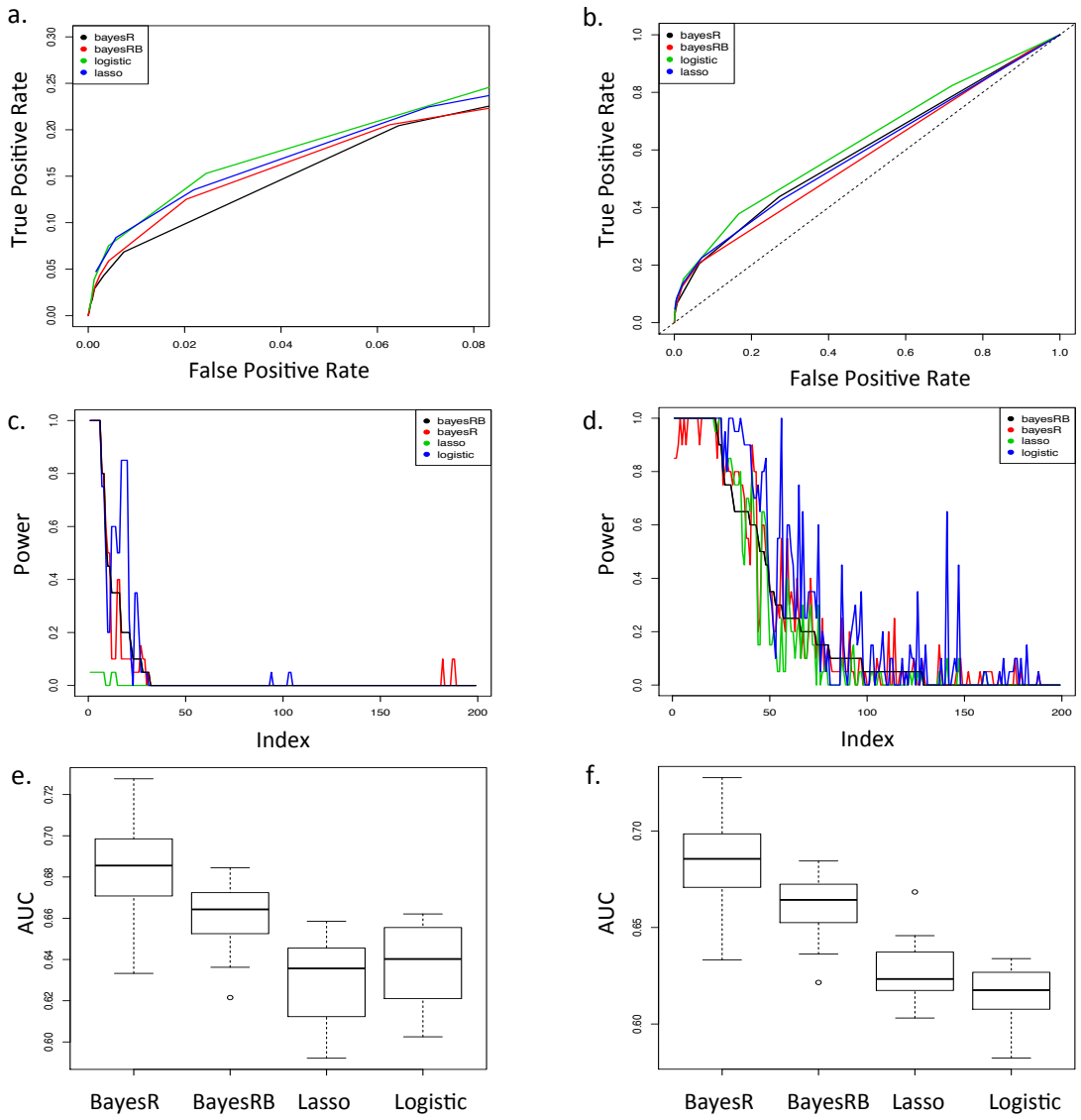


Figure 3.22: **Comparisons of BayesRB, BayesR, logistic regression and LASSO's associated SNP selection performance and risk prediction performance in the CD data sets.**

a. and **b.** True positive rate vs. false positive rate of detecting the 250kb windows containing the 201 previously reported SNPs. The black dotted line in **b.** is the diagonal line. **c.** and **d.** Power to detect the windows containing the 201 previously reported CD associated SNPs in the 20 replicates. The SNPs are sorted by a decreasing order of their BayesRB powers. **e.** and **f.** AUCs of BayesRB, BayesR, LASSO, and logistic regression in 20 replicates. In **c** and **e**, The thresholds of logistic regression and the LASSO are set under the FPR of 0.001. In **d** and **f**, The thresholds of logistic regression and the LASSO are set under the FPR of 0.05.

Table 3.2: **The contingency table of genotypes of rs11887827 and its highly correlated SNPs in cases and controls, respectively.**

a.

Controls		rs7593114		
		0	1	2
rs11887827	0	236	0	0
	1	0	1005	0
	2	0	0	984

Cases		rs7593114		
		0	1	2
rs11887827	0	128	0	0
	1	0	451	2
	2	0	115	564

b.

Controls		rs6715049		
		0	1	2
rs11887827	0	232	4	0
	1	6	996	3
	2	0	5	979

Cases		rs6715049		
		0	1	2
rs11887827	0	124	4	0
	1	0	446	7
	2	1	114	564

a The contingency table of rs11887827 and rs7593114 genotypes in cases and controls, respectively.
b The contingency table of rs11887827 and rs6715049 genotypes, in cases and controls, respectively.

Bipolar Disorder (BD) data set

For Bipolar Disorder (BD) data set, we also randomly selected one training data set out of 20 to compare the SNP effect estimation performance and the associated SNP detection performance of BayesRB, BayesR and logistic regression. Figure 3.23a shows that the BayesRB SNP effect estimates have a linear relationship of the BayesR estimates, except for two SNPs: rs4923955 and rs16957168. And the other two SNPs have much bigger BayesRB and BayesR SNP effect estimates than other SNPs. The two SNPs are rs12050604 and rs1381855. The four SNPs locate close to each others on chromosome 15. The farthest two SNPs are only 62.3kb away from each other. Among the four SNPs, only rs12050604 has a significant p value (5.08×10^{-7}) in marginal logistic regression analysis. The other three SNPs all have p value bigger than 0.05. The two green dots locating at the bottom in chromosome 15 in Figure 3.24a are rs4923955 and rs1381855, which do not have significant

p value but has top 10 SNP effect estimates of BayesRB. The two green dots locating at the bottom in chromosome 15 in Figure 3.24b are rs16957168 and rs1381855, which do not have significant p values but have top 10 SNP effect estimates of BayesR. However, when we reestimated the SNP effects using logistic regression conditional on rs12050604, both rs1381855 and rs16957168 are pumped up (Figure 3.25). Figure 3.24c and d show that with the rs12050604 conditioned on, while both rs12050604 and rs1381855 are on the top of the manhattan tower, rs4923955 is still at the bottom, and rs16957168 is halfway up the tower. Figure 3.23b and c show that before the rs12050604 is conditioned on, rs12050604, rs1381855 and rs4923955 have much larger BayesRB effect estimates than logistic regression; however, after rs12050604 is conditioned on, BayesRB has similar SNP effect estimates of rs12050604 and rs1381855 to logistic regression estimates, but it still has much larger effect estimate of rs4923955 than logistic regression. Table 3.3 shows the contingency table of rs12050604 and the other three SNPs' genotypes. While only 14 out of 2224 controls have homozygous rs16957168 mutations and heterozygous rs12050604 mutations, 374 out of 1332 cases have the same genotypes. While 193 out of 2224 controls have homozygous rs12050604 mutations and heterozygous rs4923955 mutations, only 19 out of 1332 cases have the same genotypes. The joint distributions of rs12050604 and rs1381855 genotypes are similar in cases and controls.

For the BD data set, we also compared the TPR and FPR of BayesRB, BayesR, logistic regression, and LASSO to detect the 250kb windows containing the previously reported SNPs; compared the power of the four methods of having each window containing causal SNPs identified within the 20 replicates; and compared the performance of risk prediction of the four methods on the testing data set. The 497 previously reported SNPs are obtained from GWAS catalog [Welter et al., 2014] with reported p values smaller than 5×10^{-8}. Only the SNPs reported by studies with European ancestry samples are included. Figure 3.22a shows that when the FPR is smaller than 0.08, under the same value of FPRs, logistic regression and LASSO have similar TPRs; BayesRB and BayesR have similar TPRs. The logistic regression and LASSO have higher TPR than BayesRB and BayesR under the same value of FPRs. When the FPR is 0.08, the TPR is only around 0.1 for all the methods. 3.22b shows that, all in all, the TPRs are similar for the four methods, but BayesRB has

slightly lower TPR than the other three under the same FPR. Figure 3.22c and d illustrate that using the thresholds under the FPR of both 0.001 and 0.05, the four methods have similar powers. Figure 3.22e and f show that BayesRB and BayesR generate bigger AUCs than LASSO and logistic regression. But BayesR's median value of AUCs is higher than BayesRB. BayesRB and BayesR have similar precise predictions.

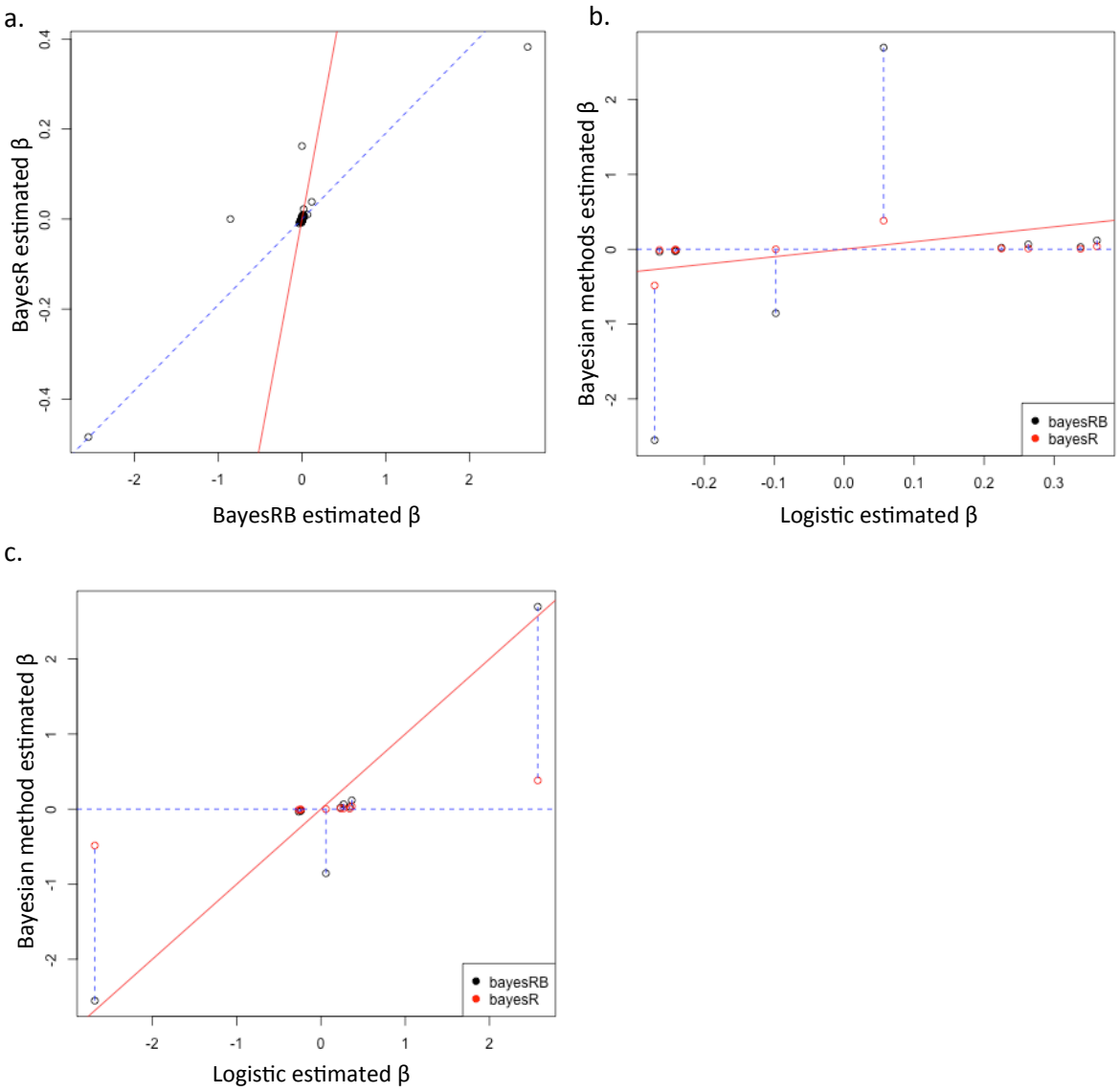


Figure 3.23: **The comparisons of SNP effect estimates by BayesRB, BayesR, and logistic regression in the BD data sets.**

The red solid lines are the diagonal lines. **a.** Compare the SNP effect estimates of BayesRB to BayesR. The blue dotted line is the fitted regression line, excluding the estimates of rs1381855 and rs4923955. **b.** Compare the marginal logistic regression estimates to BayesRB and BayesR estimates of the top 10 SNPs which have the biggest bayesRB estimates. **c.** Compare the logistic regression estimates conditional on rs12050604 to BayesRB and BayesR estimates of the top 10 SNPs which have the biggest bayesRB estimates.

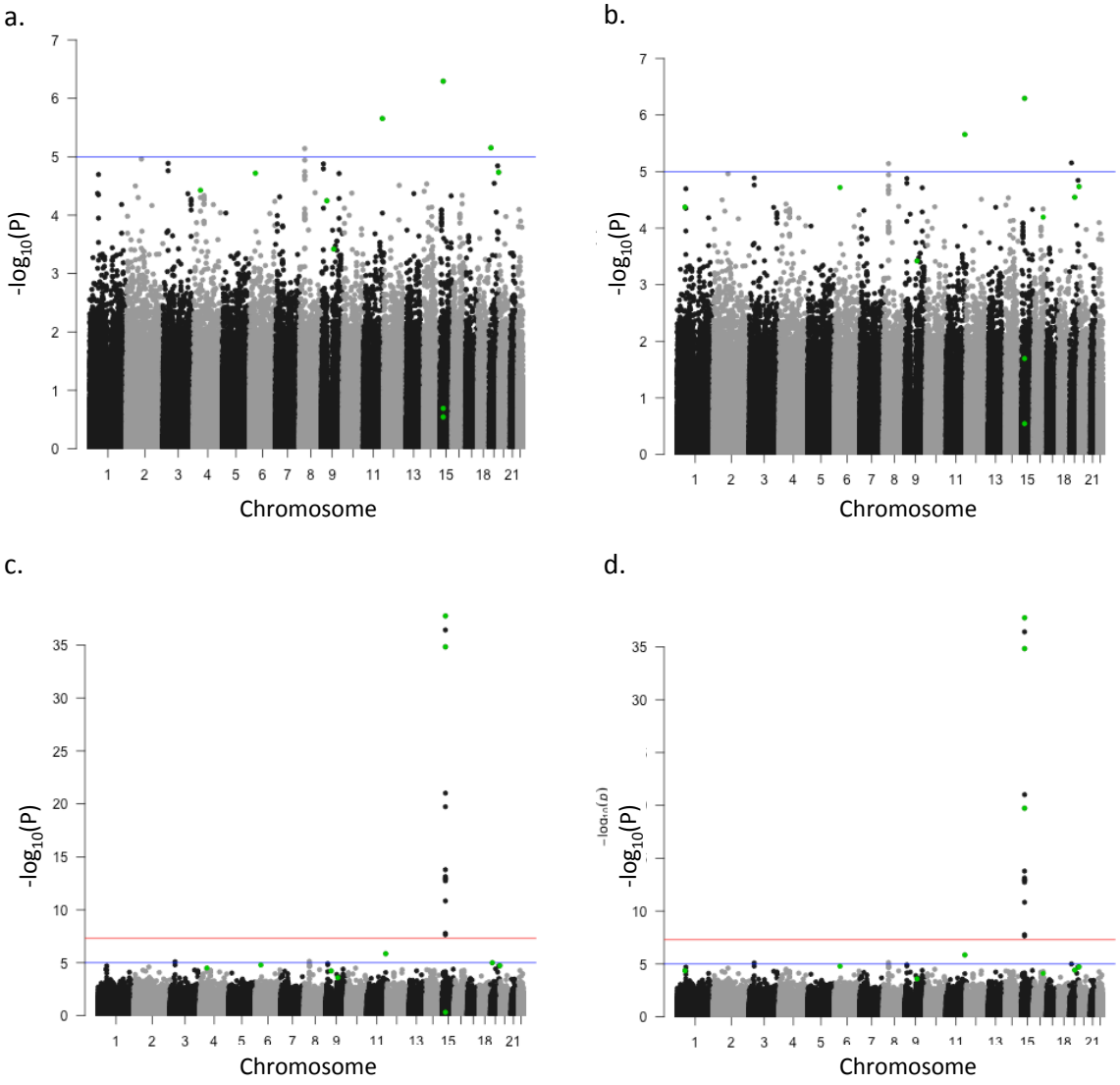


Figure 3.24: **Manhattan plots of one of the BD training data sets before and after the SNP rs12050604 is conditioned on with the top 10 BayesRB and BayesR estimated SNPs highlighted, respectively.**

a. The y axis is the $-log_{10}$ of the marginal p values of all the SNPs. The green dots indicate the 10 SNPs with the biggest BayesRB SNP effect estimates. **b.** Same manhattan plot in **a**. The green dots indicate the 10 SNPs with the biggest BayesR SNP effect estimates. **c.** The y axis is the $-log_{10}$ of the p values of all the SNPs conditioned on rs12050604. The green dots indicate the 10 SNPs with the biggest BayesRB SNP effect estimates. **d.** Same manhattan plot in **c**. The green dots indicate the 10 SNPs with the biggest BayesR SNP effect estimates.

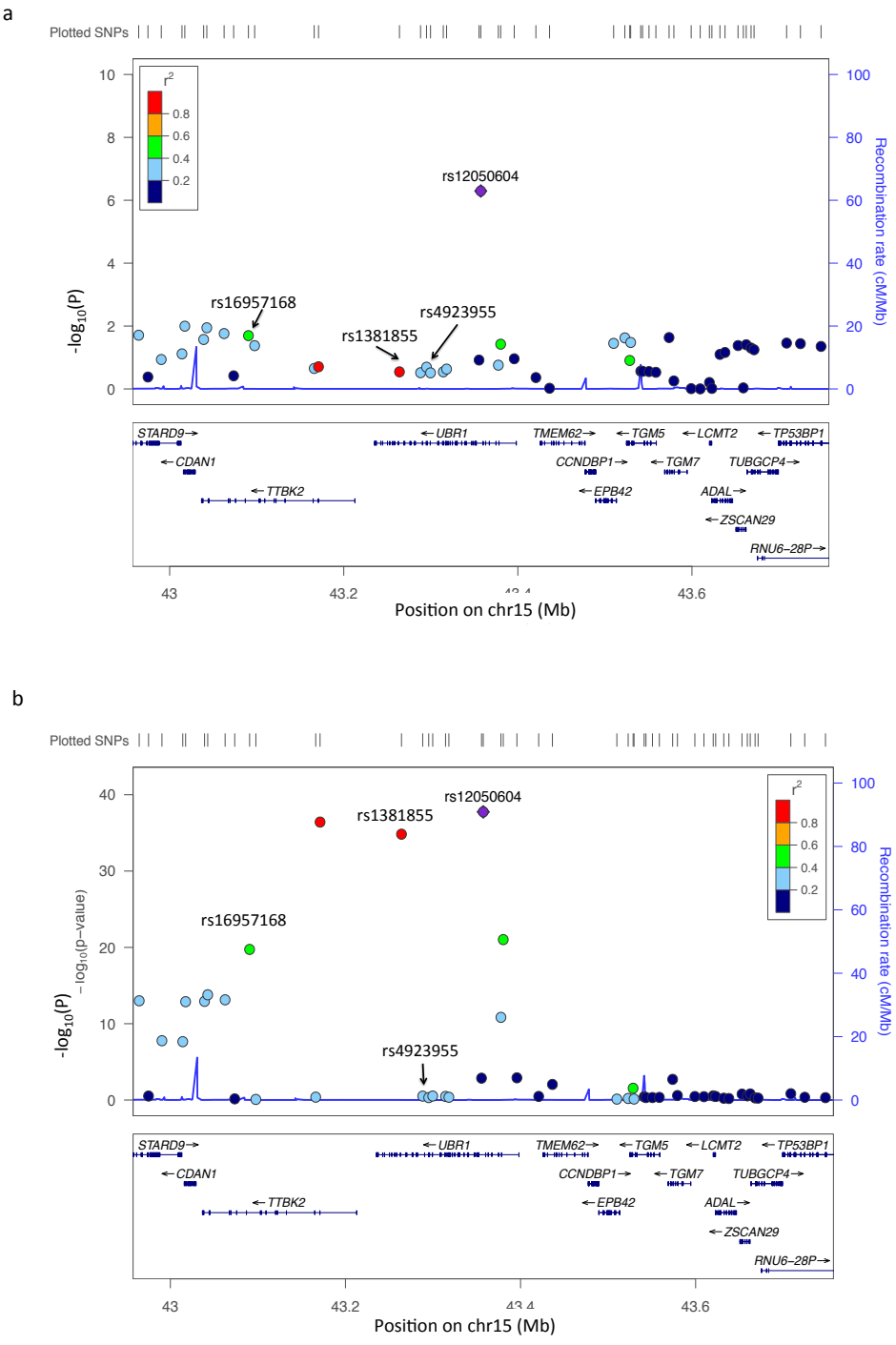


Figure 3.25: **LocusZoom plots of 400kb region on chromosome 15, centered on rs12050604.**

a. The p values are generated from marginal logistic regression analysis. **b.** The p values are generated from logistic regression analysis with rs12050604 conditioned on.

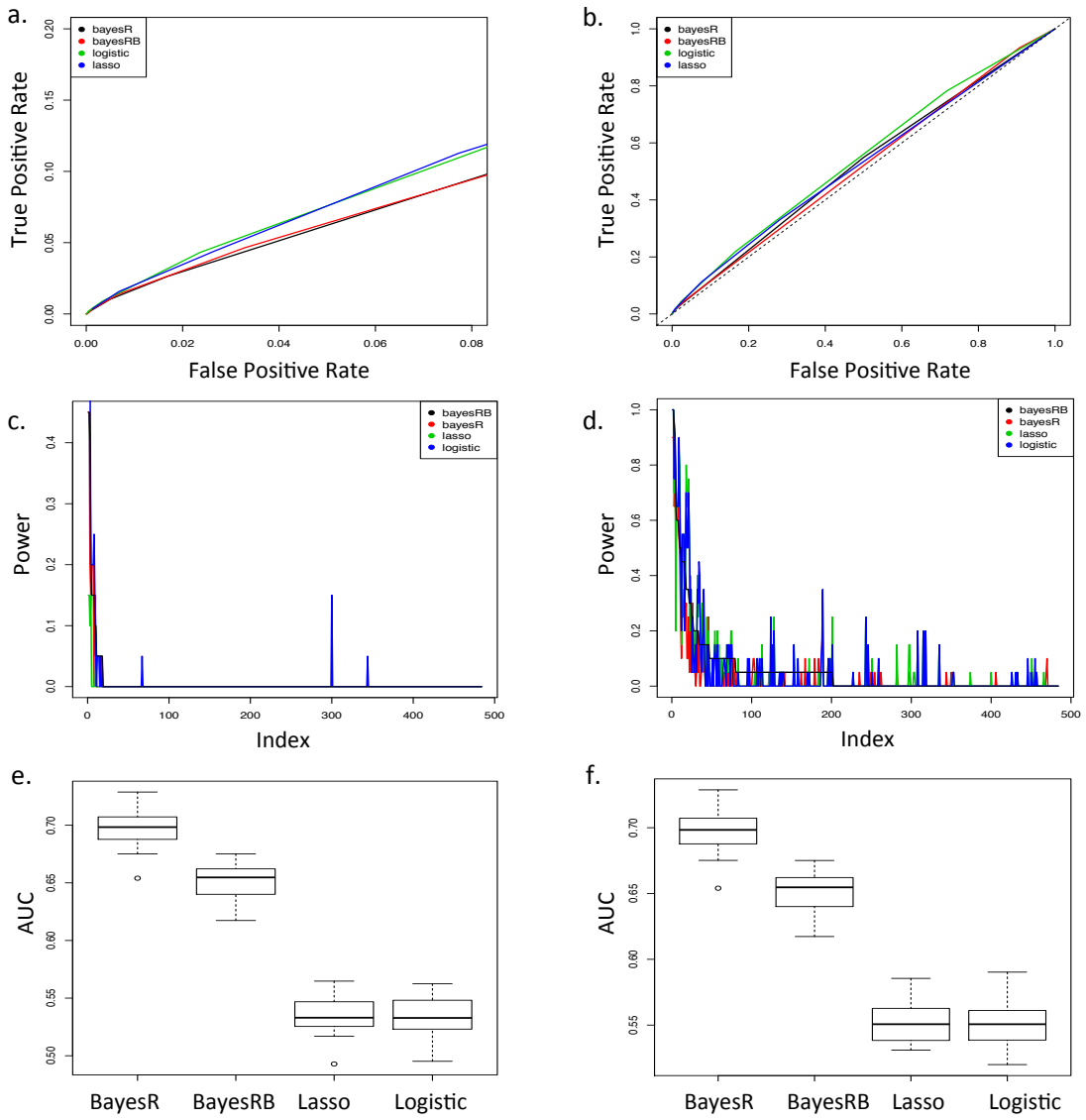


Figure 3.26: **Detection of BayesRB's associated SNP selection performance and risk prediction performance in the BD data sets.**

a. and **b.** True positive rate vs. false positive rate of detecting the 250kb windows containing the 497 previously reported SNPs. The black dotted line in **b.** is the diagonal line. **c.** and **d.** Power to detect the windows containing the 497 previously reported BD associated SNPs in the 20 replicates. The SNPs are sorted by a decreasing order of their BayesRB powers. **e.** and **f.** AUCs of BayesRB, BayesR, LASSO, and logistic regression in 20 replicates. In **c** and **e**, The thresholds of logistic regression and the LASSO are set under the FPR of 0.001. In **d** and **f**, The thresholds of logistic regression and the LASSO are set under the FPR of 0.05.

Table 3.3: **The contingency table of genotypes of rs12050604 and its highly correlated SNPs in cases and controls, respectively.**

a.

Cases		rs16957168		
		0	1	2
rs12050604	0	73	93	39
	1	2	544	374
	2	0	14	1085

Controls		rs16957168		
		0	1	2
rs12050604	0	48	64	19
	1	2	365	14
	2	0	10	810

b.

Controls		rs1381855		
		0	1	2
rs12050604	0	48	4	0
	1	6	996	3
	2	0	5	979

Cases		rs1381855		
		0	1	2
rs12050604	0	204	1	0
	1	10	908	2
	2	0	18	1081

c.

Controls		rs4923955		
		0	1	2
rs12050604	0	33	92	80
	1	4	349	567
	2	0	19	1080

Cases		rs4923955		
		0	1	2
rs12050604	0	16	65	50
	1	0	3	378
	2	0	193	627

a. The contingency table of rs12050604 and rs16957168 genotypes in cases and controls, respectively. **b.** The contingency table of rs12050604 and rs1381855 genotypes, in cases and controls, respectively. **c.** The contingency table of rs12050604 and rs4923955 genotypes, in cases and controls, respectively.

3.4 DISCUSSION

When developing our BayesRB method, we investigated two implementations. In the first, we estimated the genetic variance σ_g^2, similar to what Moser et al. [2015] did. We set a uniform prior for the σ_g^2 and estimated it using Metropolis-Hasting sampling. In the second, we fixed the σ_g^2 as Erbe et al. [2012] did. Based on our results comparing these two approaches, we decided not to use Metropolis-Hasting sampling to estimate σ_g^2 for the realistic pilot simulated data set, genome-wide simulated data sets, and the real data sets

for the following two reasons: First, a fixed σ_g^2 generates similar $\boldsymbol{\beta}$ estimates as the random σ_g^2 does. The unrealistic data set showed that when σ_g^2 is set to a fixed value bigger than 10 or is estimated using Metropolis-Hasting sampling, the BayesRB $\boldsymbol{\beta}$ estimates are almost equivalent to each other (Figure 3.6). Using the Metropolis-Hasting sampling, the variance of the proposal distribution has to be tuned manually, which has low efficiency for large data sets. So, it is reasonable to use fixed σ_g^2 instead. Second, the variance of the SNP effects of the realistic data set, genome-wide simulated data sets, and the real data sets are all very small compared to the individual variances λ. The four categories are distinguished by the four different variances of the normal distributions. Only when the variance is big enough, the four categories can be distinguished from each other (Figure 3.27a). When the variance of the SNP effects are very small compared to the λ, it is hard to tell the difference of the four categories (Figure 3.27b). In this case, σ_g^2 cannot be estimated well. Therefore, we set σ_g^2 fixed for the realistic data set of the pilot simulated data sets, genome-wide simulated data sets, and the real data sets.

When the σ_g^2 is small ($\sigma_g^2 \leq 10$), the $\boldsymbol{\beta}$ is underestimated. Both the mean and the variance of the posterior distribution of $\boldsymbol{\beta}$ is a function of σ_g^2. When the σ_g^2 is small , even the SNP is assigned to the fourth category, the $\boldsymbol{\beta}$ prior has a small variance and a zero mean and the $\boldsymbol{\beta}$ prior has a large effect on the $\boldsymbol{\beta}$ posterior distribution according to the Formula 3.14. So the BayesRB $\boldsymbol{\beta}$ tends to be underestimated. When the σ_g^2 is large enough ($\sigma_g^2 > 10$), the $\boldsymbol{\beta}$ prior does not have a large effect on the $\boldsymbol{\beta}$ posterior distribution any more. Data then drives the $\boldsymbol{\beta}$ posterior distribution.

When the σ_g^2 is large enough ($\sigma_g^2 > 10$), σ_g^2 affects the SNP categorization, thus, affects the shrinkage level of the SNP effect estimation. When the σ_g^2 is larger, the SNPs are more likely to be assigned to the categories with smaller variance and a larger proportion of SNPs are assigned to the category 1, so the BayesRB shrinks the SNP effect more.

We examined the SNP effect estimation performance of BayesRB, BayesR, and logistic regression. LASSO underestimates the SNP effects [Wu et al., 2009]. Therefore, it is not included in the comparison. The results in both genome-wide simulated data sets and the real data sets show that for the large effect SNPs, BayesRB provides similar SNP effect estimates to the logistic regression estimates, which is unbiased; while for the small effect

SNPs, BayesRB shrinks their SNP effects to zero (Figure 3.11, Figure 3.18, Figure 3.23). BayesR underestimated the SNP effects. This result is consistent with Gianola et al. [2013].

In the CD data analysis, from the BayesRB and BayesR SNP effect estimation results, we discovered that the joint distributions of rs11887827 and two of its highly correlated SNPs are quite different in cases and in controls (Table 3.2); in the BD data analysis, we discovered that the joint distributions of rs12050604 and two of its highly correlated SNPs are quite different in cases and in controls (Table 3.3). In the WTCCC study, the case data sets and the control data sets were generated separately, according to the WTCCC website. We made scatter plots using the fluorescent signal intensity data of the two alleles of rs11887827 and rs12050604 for each individual, as well as their highly correlated SNPs. Intensity plots of both rs11887827 and rs12050604 show four clusters instead of three (Figure 3.28), which may be caused by duplications [Kumasaka et al., 2011]. All the highly correlated SNPs show a normal pattern of three clusters (data not shown). Figure 3.28 show that, for both rs11887827 and rs12050604, different decisions were made in controls and cases when calling the genotypes in cluster 2. For both SNPs, controls in cluster 2 are treated as having 1/0 genotypes, but cases in cluster 2 are treated as having 1/1 genotypes. Thus, different decisions on how to call the genotypes in cluster 2 lead to the different joint distributions of genotypes in cases and controls.

The observation discussed above in the CD and BD data sets are due to batch effects, however, similar observation could be due to some real signals. In reality, it is possible that some markers become significant after conditioning on other markers. It is also possible that some markers are no longer significant after conditioning on other markers. In GWAS, people often conduct an additional conditional analysis to further investigate the markers conditional on the significant ones. If the first marker itself is not significant (like rs11887827 has p value $< 10^{-5}$ in CD) in marginal logistic regression, then this marker will not be conditioned on, and thus no other markers can be discovered. BayesRB estimates the SNP effects simultaneously and makes up the defects of the conventional marginal logistic regression method. It can save effort to conduct the conditional analysis after discovering the significant markers. It may not fail to detect the jointly significant markers, nor mistakenly include the markers in the model that are significant only because they are in LD with more

significant markers. In addition, BayesRB can help with quality control. For example, in our analyses, it discovered two SNPs affected by batch effects. These problematic SNPs should be deleted from the data sets. All in all, the above reasons illustrates the potential value of the BayesRB approach over the conventional marginal logistic regression method.

We examined the associated SNP selection abilities of BayesRB, BayesR, LASSO, and logistic regression, using both genome-wide simulated data sets (Figure 3.13) and the real data sets (Figure 3.22a,b and Figure 3.26a,b). In the genome-wide simulated data sets, we set the causal SNPs at the beginning of the simulation, therefore, the TPR to detect big effect causal SNPs can be as large as 1; and the TPR to detect medium effect causal SNPs can also be larger than 0.6, when the FPR is 0.08. But in the real data sets, it is unknown which SNPs are the true causal SNPs. So we calculated the TPR and FPR of detecting the previously reported SNPs. The TPR to detect the previously reported CD and BD SNPs are as low as around 0.2 and 0.1 respectively, when the FPR is 0.08. In addition, in the genome-wide simulated study, the proportion of windows containing causal SNPs being discovered in at least one replicates is much bigger than that in the real data study. The low TPRs of detecting the previously reported SNPs within 250kb regions, and the low power of having the windows containing the previously reported SNPs being detected in CD and BD data sets are because of the following reasons: First, for the real data sets, we calculated TPR and FPR using all of the previously reported SNPs regardless of how big their effect size estimates turned out to be in the BayesRB analysis. So, it is reasonable that the TRP is smaller than only using the big and medium effect SNPs as we did in genome-wide simulated data sets. Second, some previously reported associated SNPs are discovered from studies with large sample sizes ($> 20,000$). But the sample sizes in the CD and BD training data sets are only 3,485 and 3,556, respectively. Given the relatively small sample sizes in CD and BD data sets, only SNPs having relatively big odds ratios are detectable. Third, even if some SNPs that are discovered to be associated with CD and BD from data sets with similar sample sizes to those of the WTCCC CD and BD data sets, they may not be detectable using the WTCCC CD and BD data sets. Fourth, some SNPs are reported to be associated with CD or BD, but they may not be. The SNPs may be discovered due to some false positive signals, and thus cannot be replicated using other data sets.

Based on the results of TPR, FPR and power of identifying windows containing causal SNPs in the genome-wide simulated data sets (Figure 3.13), and identifying windows containing previously reported SNPs in the real data sets (Figure 3.22, Figure 3.26), we concluded that BayesRB does not perform better than the other three methods. However, BayesRB is still a promising method for identifying associated SNPs in real studies and guiding the direction of future studies. The BayesRB associated SNPs are those with proportion of iterations that are assigned to the category 2, 3 or 4 (assignment proportion) bigger than the threshold. Figure 3.7 and Figure 3.12 shows that the SNPs with bigger logistic regression estimates have bigger assignment proportions, and thus are likely to be selected as the BayesRB associated SNPs. This shows BayesRB's good performance in terms of its ability to select associated SNPs. Figure 3.7 and Figure 3.12 are slightly different: First, many more SNPs in Figure 3.7 have big assignment proportions than that in Figure 3.12. This is because Figure 3.7 is based on the unrealistic data sets, which contains SNPs with bigger effects than the reality; while Figure 3.12 is based on the genome-wide simulated data sets, which contains the SNPs with effects similar to the real data sets. Second, in Figure 3.12, some SNPs have logistic estimates as big as around 0.4, but have assignment proportions close to 0; while in Figure 3.7, all the big logistic regression estimated SNPs have big assignment proportions. This is due to the reason that in the unrealistic data sets, all the SNPs are independent of each other; while in the genome-wide simulated data sets, the SNPs are not fully independent. The SNPs having big logistic regression estimates but small proportions in the genome-wide simulated data sets may be correlated to some other more significant SNPs, thus, they are not selected in the model. Table 3.4 shows the detailed information of the top 10 SNPs with the biggest assignment proportions in the CD data set, excluding the two SNPs on chromosome 2 affected by the batch effects. 8 out of 10 SNPs locate within the 500 kb windows of the previously reported CD associated SNPs. Table 3.5 shows the detailed information of the top 10 SNPs with the biggest assignment proportions in the BD data set, excluding the three SNPs on chromosome 15 affected by the batch effects. 7 out of 10 SNPs locate within the 500kb windows of the previously reported BD associated SNPs. While BayesRB can be applied to identify the associated SNPs in the real studies, it also suggests that, in the future, more studies need to be conducted to assess the two SNPs that

were not previously reported to be associated with CD and the three SNPs that were not previously reported to be associated with BD.

We also measured the prediction performance of BayesRB, BayesR, LASSO, and logistic regression using AUCs. BayesR has the best prediction performance than the other three methods, followed by BayesRB. In the genome-wide simulated data sets, which are not affected by the batch effects, when the FPRs of logistic regression and LASSO are 0.001, BayesR and BayesRB have slightly bigger AUCs than logistic regression and LASSO (Figure 3.16). In the CD and BD data sets, BayesR and BayesRB have much larger AUCs than logistic regression and LASSO (Figure 3.22e,f and Figure 3.26e,f). One possible reason is that in CD and BD data sets, the BayesRB and BayesR models not only contain rs11887827 and rs12050604, respectively, but also contain one of their correlated SNPs. Containing rs11887827 and rs12050604, respectively, in the models alone, as logistic regression does, does not fully account for the association signal in CD and BD. Therefore, BayesRB and BayesR have much better risk prediction performance than logistic regression and LASSO. The AUCs should be re-compared after more thorough quality control to identify and delete the SNPs affected by batch effects. But if the data truly contains some jointly significant markers, BayesRB and BayesR are expected to show better prediction performance than the conventional approaches.

When calculating the power of identifying the windows that containing the causal SNPs and calculating the AUCs of logistic regression and LASSO, we use the thresholds with the FPR equals to 0.001. A multiple comparison adjusted FPR is more appropriate ($\sim 10^{-6}$), but we did not use the multiple comparison adjusted FPR in this study. The reason is justified below. When the FPR is constrained to 0.001, the power of identifying most of the windows containing the causal SNPs are small. If the FPR were smaller, less SNPs would be identified as associated SNPs. Thus, the power of identifying the windows that containing the causal SNPs will be even smaller. In this case, all the four methods performs equally poor. When FPR is constrained to 0.001, four methods' associated SNP selection performance can be better compared with each other. Considering the consistency of the dissertation, when calculating the AUCs of logistic regression and LASSO, we also set the thresholds using a FPR of 0.001.

It is recommended to use BayesRB instead of BayesR for dichotomous outcome data, even though BayesR has slightly better risk prediction performance than BayesRB. First, the SNP odds ratios can be estimated by BayesRB, but cannot be estimated by BayesR. BayesRB is based on a logistic regression model, while BayesR is based on an ordinary linear model. So, the estimated SNP odds ratios can be calculated by taking the log of the estimated SNP effects of BayesRB, but cannot be calculated by the estimated SNP effect of BayesR. Second, BayesRB can predict the risk of getting disease, but BayesR cannot. The predicted outcome of BayesR is hard to explain. It is not the risk of getting disease, since the estimated outcome can be bigger than 1.

When applying BayesRB and BayesR, convergence diagnostics should be conducted. Before applying the BayesRB or BayesR method, it is necessary to try different total numbers of MCMC loops, as well as numbers of burn-in loops, and select the ones where all the parameters mix well. But Moser et al. [2015] failed to do so. They used the default setting of the total number of MCMC loops and the number of burn-in loops of BayesR and they did not make diagnostic plots of their parameters, so it is not clear whether the parameters mix well in their MCMC process. In our studies, we conducted the convergence diagnostics in pilot simulated data analysis, genome-wide simulated data analysis and the real data analysis. Although, due to the limited computational time, BayesRB was run for less loops than needed to make all the parameters mix well, the convergence diagnostics still provide an idea of how well the parameters mixed.

To summarize, for SNP effect estimation, BayesRB has similar estimates to logistic regression for big effect SNPs, and shows BayesR's sparseness characteristic for small effect SNPs. It makes up the defects of the conventional marginal logistic regression method and estimates the SNP effects taking account of the other SNPs. It also has better risk prediction performance than logistic regression and LASSO. Although BayesRB's risk prediction performance is not better than BayesR and it does not have better associated SNP selection performance, BayesRB is still a promising method to use for dichotomous outcome data.

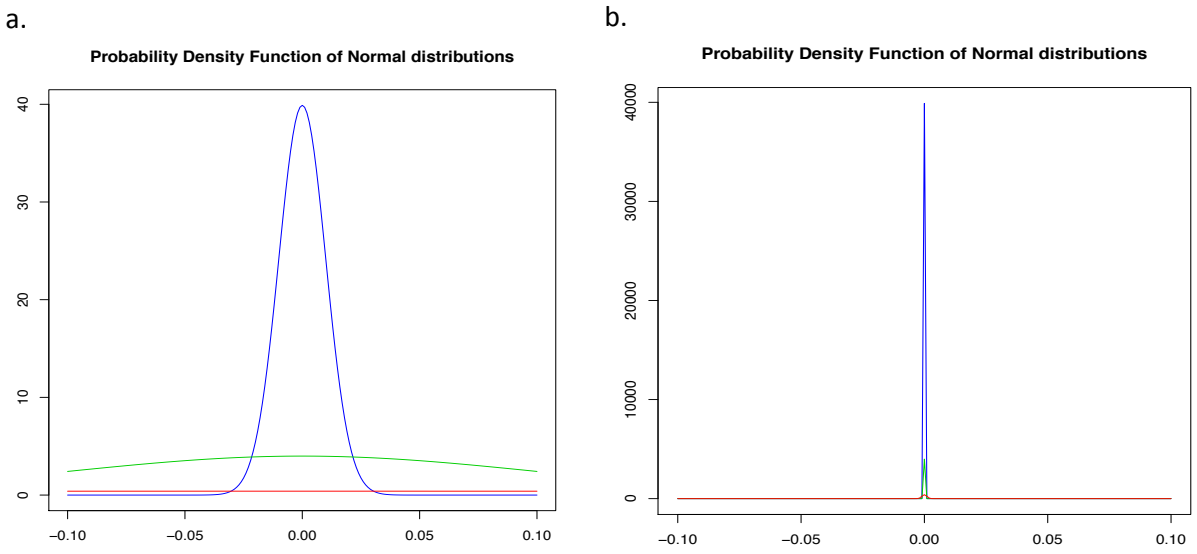


Figure 3.27: **The probability density functions of the normal distributions.**

The blue curves represent the $N(0, C_2\sigma_g^2)$; the green curves represent the $N(0, C_3\sigma_g^2)$; and the red curves represent the $N(0, C_4\sigma_g^2)$, where $(C_2, C_3, C_4) = (10^{-4}, 10^{-3}, 10^{-2})$, respectively. **a.** $\sigma_g^2 = 100$. **b.** $\sigma_g^2 = 0.1$.

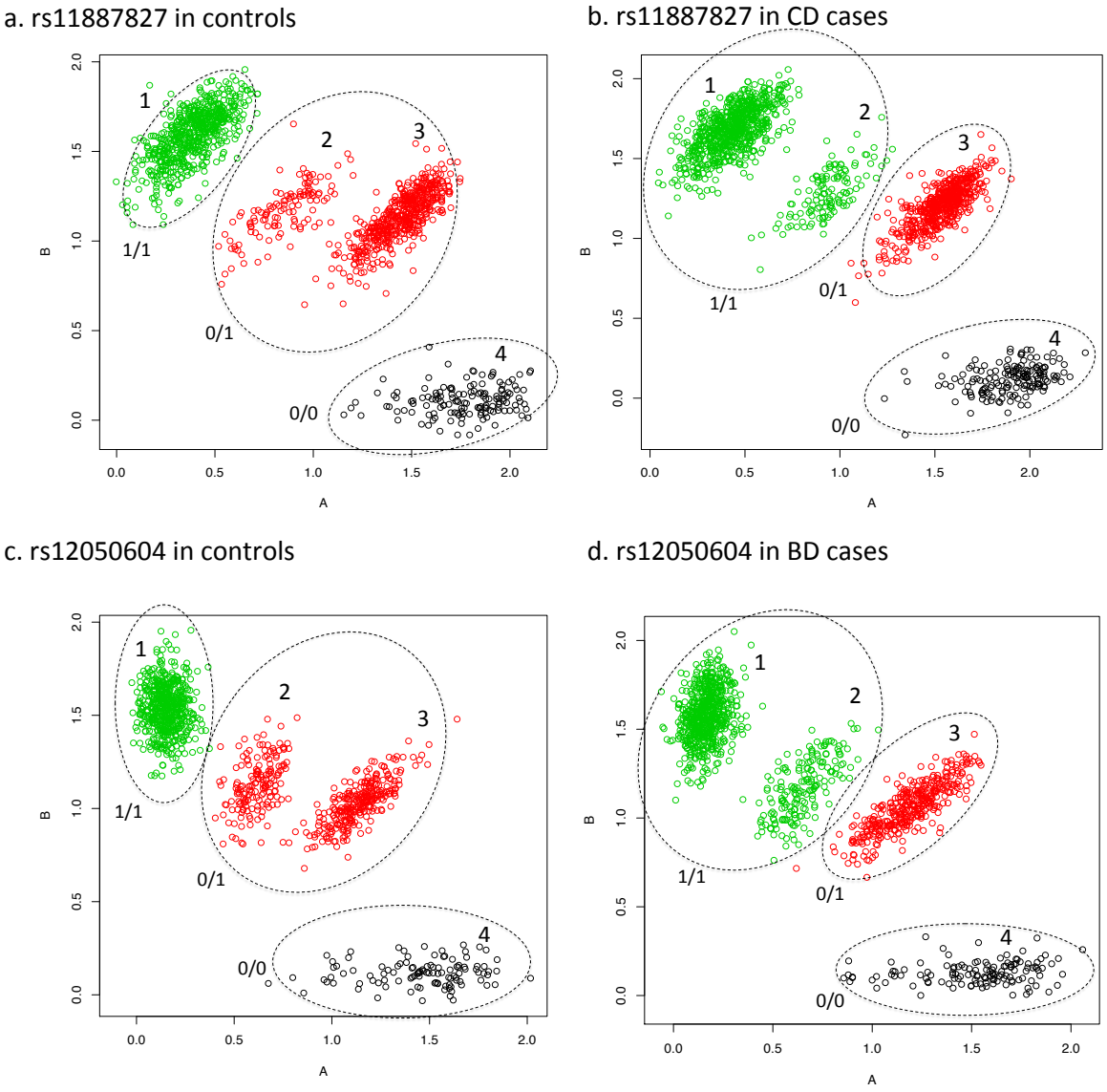


Figure 3.28: **Fluorescent signal intensity plots for rs11887827 and rs12050604 in controls and cases, respectively.**

The x-axis corresponds to the allele A and the y-axis corresponds to the allele B, respectively. The black dots indicate the individuals with coded genotypes of 0/0. The red dots indicate the individuals with coded genotypes of 0/1. The green dots indicate the individuals with coded genotypes of 1/1. The numbers "1", "2", "3" and "4" in the plots indicate the four clusters.

Table 3.4: **Detailed information of the top 10 SNPs with the biggest proportions to be assigned to the category 2, 3 or 4, excluding the two SNPs on chromosome 2 affected by the batch effects.**

SNP	CHR	BP	Proportion	BayesRB	logistic	PR SNPs	Consequence	Genes
rs7515029	1	67308393	0.54	-0.340	-0.780	rs2064689	upstream	C1orf141
rs11209033	1	67456521	0.522	0.1458	0.307	rs10789230	downstream	RP11-131O15.2
rs3828309	2	233962410	0.998	-0.304	-0.321	rs3828309	intron variant	ATG16L1 SCARNA5
rs903228	2	53603700	0.460	0.222	0.518	-	intergenic variant	-
rs6596075	5	131770127	0.303	-0.096	-0.343	rs6596075	upstream	C5orf56
rs17234657	5	40437266	0.993	0.392	0.460	rs17234657	regulatory region variant	-
rs4075496	7	69614100	0.542	-0.210	-0.361	-	intron variant	AUTS2
rs2066843	16	49302700	0.765	0.223	0.325	rs2066844	synonymous variant	NOD2
rs7186163	16	49244058	0.705	0.176	0.275	rs17221417	downstream	NKD1
rs2542151	18	12769947	0.575	0.172	0.301	rs2542151	upstream	RP11-973H7.1

The column name "CHR" indicates the chromosome number; "BP" indicates the SNP position; "Proportion" indicates the proportion of the iterations that the SNPs are assigned into the category 2, 3 or 4; "BayesRB" indicates the BayesRB estimated SNP effects; "logistic" indicates the logistic estimated SNP effects; "PR SNPs" indicates the SNPs that are previously reported to be associated with CD, locating within the 500kb windows of the top 10 SNPs. If there are multiple such SNPs, only the ones closest to the top 10 SNPs are listed. "Consequence" indicates the function of the SNPs. "downstream" indicates the downstream gene variants. "upstream" indicates the upstream gene variants. "Genes" indicates the genes that the SNPs locate on or close to.

Table 3.5: **Detailed information of the top 10 SNPs with the biggest proportions to be assigned to the category 2, 3 or 4, excluding the three SNPs on chromosome 15 affected by the batch effects.**

SNP	CHR	BP	Proportion	BayesRB	logistic	PR SNPs	Consequence	Genes
rs6447534	4	46978809	0.093	0.030	0.338	-	Intron variant	GABRB1
rs17639988	6	37788570	0.320	0.117	0.361	-	intergenic variant	MDGA1
rs10971738	9	33845138	0.102	-0.025	-0.242	rs216345	Intron variant	WHRN
rs10982256	9	114340388	0.08	-0.017	-0.211	rs10982256	Intron variant	WHRN
rs11138278	9	79363092	0.078	0.021	0.225	rs914715	intron variant	RP11-375O18.2
rs356242	11	126989854	0.268	0.065	0.263	rs11220082 rs548181	intergenic variant	-
rs1499318	13	67981591	0.090	0.017	0.201	-	intergenic variant	-
rs12979795	19	12578847	0.098	-0.021	-0.240	rs7247513	intergenic variant	ZNF490
rs10853835	19	56720762	0.080	-0.020	-0.267	rs62110082	intergenic variant	SIGLEC6
rs7248493	19	63402920	0.135	-0.034	-0.264	rs7247513	intergenic variant	ZNF274

The column name "CHR" indicates the chromosome number; "BP" indicates the SNP position; "Proportion" indicates the proportion of the iterations that the SNPs are assigned into the category 2, 3 or 4; "BayesRB" indicates the BayesRB estimated SNP effects; "logistic" indicates the logistic estimated SNP effects; "PR SNPs" indicates the SNPs that are previously reported to be associated with BD, locating within the 500kb windows of the top 10 SNPs. If there are multiple such SNPs, only the ones closest to the top 10 SNPs are listed. "Consequence" indicates the function of the SNPs. "Genes" indicates the genes that the SNPs locate on or close to.

3.5 ACKNOWLEDGMENT

This study makes use of data generated by the Wellcome Trust Case-Control Consortium. A full list of the investigators who contributed to the generation of the data is available from www.wtccc.org.uk. Funding for the project was provided by the Wellcome Trust under award 076113, 085475 and 090355

4.0 AIM 3. A SIMULATION BASED "UNSEEN SPECIES" METHOD TO ESITMATE THE TOTAL NUMBER OF DISEASE GENES IN A RECESSIVE FORWARD GENETIC SCREENING STUDY (RFGSS)

4.1 BACKGROUND AND SIGNIFICANCE

Recessive forward genetic screening study (RFGSS) is widely conducted for disease mutation detection. Figure 4.1 shows the RFGSS process. In RFGSS, animals of generation 0 (G0) are mutagenized with ethylnitrosourea (ENU) and mated with normal animals to generate mice of generation 1 (G1). G1 are mated with normal animals to generate mice of generation 2 (G2). G2 daughters are selected at random to backcross to their G1 fathers to generate generation 3 (G3) mutants with recessive mutations causing the disease. The G3 mutants with ENU induced homozygous alleles are identified by the observation of having the disease, and subsequent mapping of the allele within the genome reveals genes that are associated with the disease. Under the RFGSS scenario, failing to detect a disease gene may due to the following three reasons: First, the disease genes are not mutagenized by ENU; second, the G3 animals do not have homozygous mutations in the disease genes; third, people fail to detect the homozygous mutations in the G3 animals.

It is very necessary to estimate the total number of disease genes in a RFGSS. RFGSS assesses disease etiology by discovering the disease genes efficiently. Estimation of the total number of disease genes in a RFGSS sheds light on what percentage of genes have been detected so far. It will guide the gene screening strategy, which allows for the detection of all the disease genes and allows a better understanding of the disease. Therefore, it is necessary to develop a method to estimate the total number of disease genes in a RFGSS.

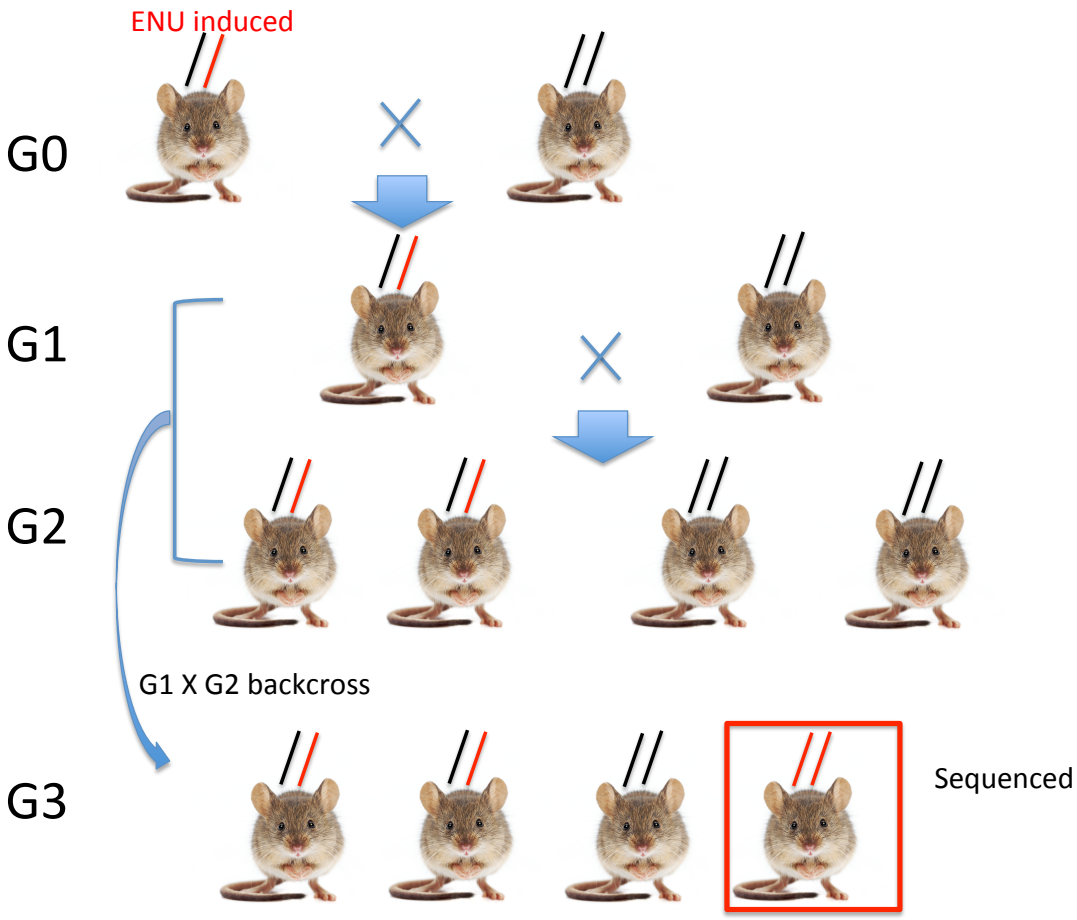


Figure 4.1: **Recessive forward genetic screening study process.**

It is hard to estimate the total number of disease genes based on the RFGSS process directly, because of the following reasons: First, the distribution of the number of mutations in G0 animals induced by ENU mutagenesis is unknown; second, the probability of a homozygous mutation in a diseased G3 animal being detected is unknown. In addition, since only the diseased G3 animals with at least one disease mutation are screened and only the genes with at least one mutations are observed, the observed numbers of mutations in the observed genes of the screened animals are not independent of each other. Therefore, the likelihood function is hard to derive.

In previous studies, people conducted parametric methods on the total number estimation based on the observation. In a large-scale screening study for loss-of-function mutants, Pollock and Larkin [2004] proposed a parametric method using Bayesian and maximum-likelihood methods, based on the location and distribution of the detected disease mutations. Luliana et al. [2009] used a parametric beta-binomial model to estimate the total number of unseen variants in the human genome.

Sampling genes is analogous to sampling species in ecology, therefore, the ideas from the "unseen species" methods, which estimate the total species extant in a geographic region, can be borrowed. The complicated process of obtaining the observed homozygous mutations in G3 animals in the RFGSS can be regarded as the process of animal catching in the "unseen species" problem. Some studies have applied this idea to answer the genetic questions before, although not applied to RFGSS. Chao and Lee [1992] developed a sample coverage based nonparametric method to estimate the total number of species. Sanders et al. [2012] and Zaidi et al. [2013] applied the method to estimate the number of genes associated with autism spectrum disorders risk in a parent-sibling-pair study and to estimate the number of genes in which de novo mutations contribute to congenital heart disease in a parent-offspring-trio study, respectively. The sampling coverage based method uses the observed frequency and the number of detected disease-related genes (or species) to infer the total number of disease-related genes in the population, including those yet to be observed. When applying the sample coverage based method, both of the studies assumed the coefficient of variation of the probability that one or more mutations falls in each gene equals to 1 without sufficient justification. We used the γ to represent the coefficient of variation of the probability that one or more mutations falls in each gene. Chao [1992] proposed a γ estimator. However, it underestimates the total number of species [Chao and Lee, 1992]. Some other nonparametric methods are also developed to solve the "unseen species" problems. For example, Burnham and Overton [1978; 1979] proposed a jackknife estimator, Chao and Bunge [2002] proposed an estimator from Poisson-Gamma model, Norris and Pollock [1996; 1998] proposed a nonparametric maximum likelihood estimator, Wang and Lindsay [2005; 2008] proposed a penalized nonparametric maximum likelihood estimator.

Here, on the basis of Chao and Lee's sample coverage based method, we proposed a simulation-based approach, which uses the idea of self consistency of Chao and Lee's method. It is a nonparametric method and can be applied to estimate the total number of disease genes in the RFGSS.

4.2 METHODS

4.2.1 Sample coverage based approach with simulated parameters

The sample coverage C is defined to be the sum of the probabilities of the genes being observed [Chao and Lee, 1992].

$$C = \sum_{i=1}^{N} p_i I[X_i > 0],$$

where N is the total number of disease genes; p_i is the probability that a mutation falls into the ith gene. $I[.]$ is an indicator function. X_i is the number of observed mutations in gene i. Good [1953] proposed an estimator of C:

$$\hat{C} = 1 - f_1/n,$$

where f_1 is the number of genes that have 1 mutation in the sample, and n is the number of observed disease mutations. We applied the formula Chao and Lee proposed for the total number of genes (species) estimation ($\hat{N}$):

$$\hat{N} = \frac{D}{\hat{C}} + \frac{n(1-\hat{C})}{\hat{C}}\gamma^2,$$

where $D = \sum f_i$, which is the observed total number of genes. f_i is the number of genes that have i mutation in the sample. $\gamma = [\sum_i (p_i - \bar{p})^2/N]^{1/2}/\bar{p}$, which is the coefficient of variation of p_i, $i = 1...N$. We assumed a constant ENU induced mutation rate across all genes, then $p_i = l_i / \sum_{i=1}^{N} l_i$, where l_i is the length of gene i. However, only the observed genes have p_i's calculated. γ cannot be directly calculated. Chao and Lee proposed an estimator ($\tilde{\gamma}$) for γ:

$$\tilde{\gamma}^2 = max\left\{ max\{\hat{N}_1 \sum_{i=1}^{n} i(i-1)f_i/[n(n-1)] - 1, 0\} \cdot \{1 + n(1-\hat{C}) \sum_{i=1}^{n} i(i-1)f_i/[n(n-1)\hat{C}]\}, 0 \right\}$$

Instead of using the $\tilde{\gamma}$ to calculate $\hat{N}$, we proposed a simulation-based method using the self-consistency idea in Chao and Lee's method to infer the likely range of possible values of total number of disease genes while taking gene lengths into account. We assumed the total number of disease genes is M, with M in the range of 200 to 3000. Although it is arbitrary to select 3000 as the upper boundary, it is reasonable. Because in reality, it is unlikely to have more than 3000 genes contributing to a disease. Then, we randomly sampled $M - n$ genes with their lengths known, based on the probabilities of having zero observation on the unobserved genes. Combining the observed gene lengths and the sampled gene lengths, a temporary $\hat{\gamma_1}$ can be calculated, thus the corresponding temporary total number of disease genes ($\hat{M_1}$) can be calculated. We set m as the M closest to the corresponding $\hat{M_1}$. Then we repeated the process for T times and obtain the median of the T m's, which is the estimate of the total number of disease genes ($\hat{N}$). The 5^{th} and the 95^{th} quantiles of the m construct an quantile interval for the $\hat{N}$. Considering the computational time, $T = 100$ in our study (Figure 4.2).

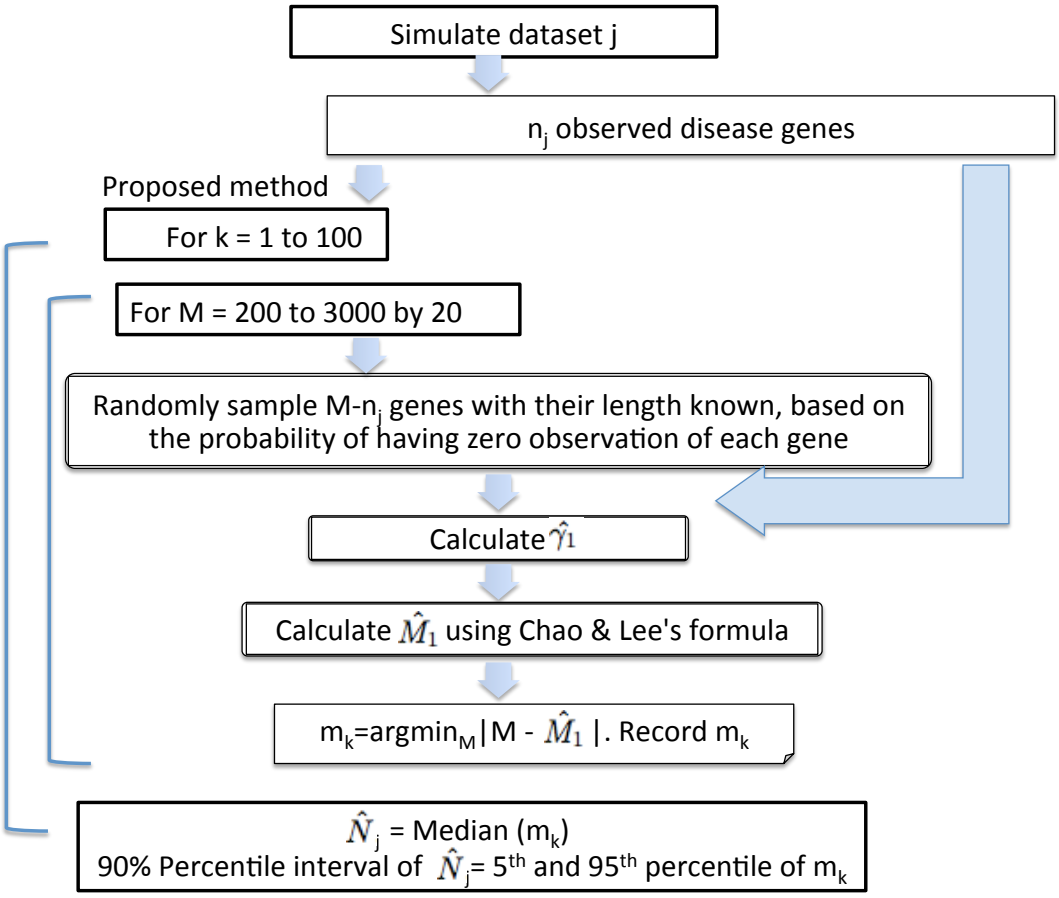


Figure 4.2: **Simulation-based "unseen species" method.**

In the process of randomly sampling $M - n$ genes based on the probabilities of having zero observation of the true disease genes, we calculated the approximated probabilities instead of the exact ones. The approximated probabilities are calculated under a thinned poisson model with the poisson parameters of ν, where ν = gene length $\times$ mutation rate $\times$ detection rate $\times$ inherited rate. Inherited rate is the probability that, if a G0 has a disease mutation, at least one of its G3 has at least one homozygous disease mutation. The exact probabilities of having zero observation on the true disease genes are hard to calculate, as they do not follow any known distribution. Thus, we used a simulation approach to access the exact probabilities of having zero observation on the true disease genes. In the simulation, we conducted the RFGSS process for 1000 times. We recorded the proportion

of times each disease gene is not observed. Figure 4.3 shows that all the genes' calculated approximated probabilities are close to their proportions of times that they are not observed in the simulation. Thus, it is appropriate to use the calculated approximated probabilities in the process of randomly sampling $M - n$ genes.

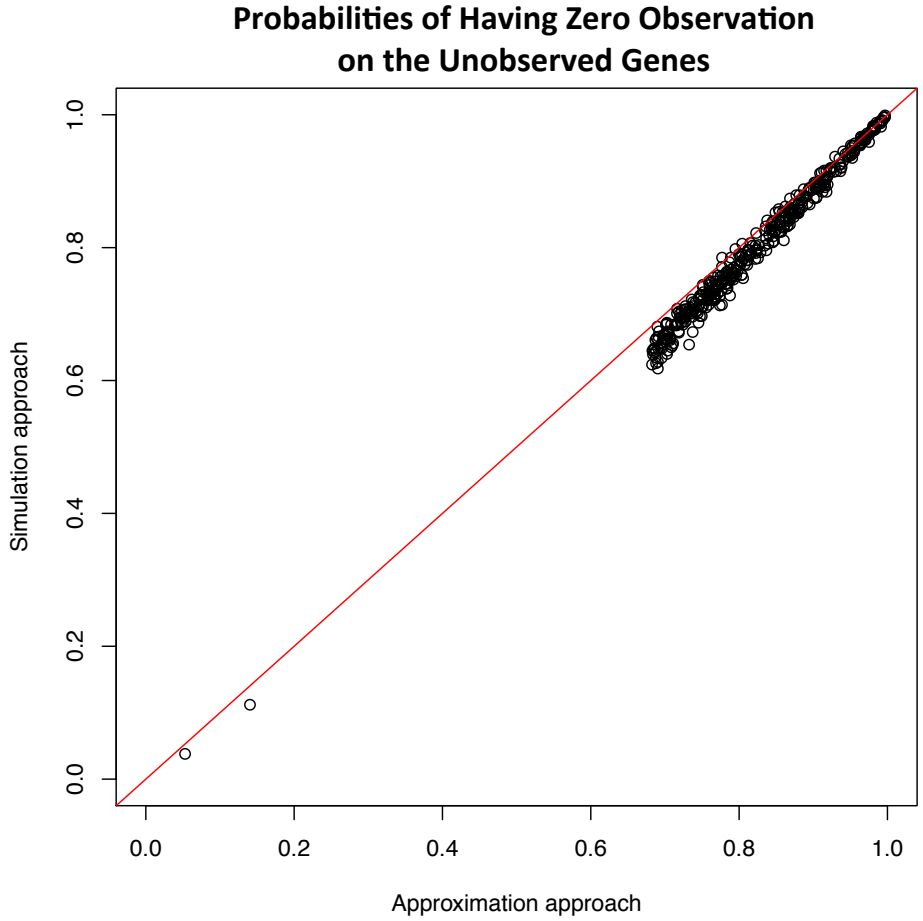


Figure 4.3: **A comparison of the simulation approach and the approximation approach when calculating the probabilities of having zero observation on the unobserved genes.**

4.2.2 Simulation study

We simulated a RFGSS process based on the information from a real study, which is a mice RFGSS of congenital heart disease (CHD) [Li et al., 2015]. In this study, the investigators

ENU-mutagenized 2,651 G0 mice with an estimated mutation rate of 5.5×10^{-7} per base. They did ultrasound scanning of the G3 fetuses from 2,651 mutant lines and recovered 218 CHD mutant lines. Mutation recovery was conducted with whole-exome sequencing analysis of a single mutant from 113 mutant lines among the 218 CHD mutant lines. They discovered 91 recessive CHD mutations in 61 genes. 90 mutations were in highly conserved amino acids. One mutation was suspected to be inherited from the ancestors of the mouse family. In the simulation, we assumed the number of mutations on the genome follows a Poisson distribution. Since 90 mutations were discovered in 113 mutant lines, we set a mutation detection rate of 90/113. We obtained the coding region length of all the genes in the mice genome from NCBI.

According to the real RFGSS mentioned above, 2,651 G0 mice should be simulated. Based on the mice family structure, the number of mutations on each CHD gene in each mice generation should be recorded and 113 G3 mice from those with more than one homozygous mutation on the CHD genes should be selected. However, this process is less efficient. The following is a more efficient process: we first simulated 800 G0 mice receiving ENU mutagenesis with a mutation rate of 5.5×10^{-7} per base. The number of mutations in each CHD gene follows a Poisson distribution with the parameter of mutation rate times the coding region length. The chance each mutation passes to G1 mice is 0.5. Then, we used the 137 G1 mice family structure information provided by Li et al. [2015] to continue the simulation study. We sampled the number of G2 mice that each G1 mouse mates with with replacement from the family structures. We also sample the number of G3 mice generated from each G1 father and each G2 mother based on the family structures. The probability each mutation passed from G1 to G2 is 0.5. And the probability a G3 mouse has a homozygous mutation is 0.25 if both of its G1 father and G2 mother have the mutation. If in a G0 family there are more than one G3 mice having more than one CHD homozygous mutations, only one such G3 mouse in the family is selected randomly. If in a G0 family only one G3 mouse having more than one CHD homozygous mutations, the G3 mouse is selected. If the number of selected G3 mice is bigger than 113, then we randomly select 113 G3 mice to do the following analysis. If the number of selected G3 mice is smaller than 113, we simulate 100 more G0 mice involving the above RFGSS process and repeat the above process until

the number of G3 mice with more than one CHD homozygous mutation is bigger than 113. The reason we simulate 800 G0 mice receiving ENU mutagenesis at the beginning is that 800 G0 mice usually generate selected G3 mice slightly more than 113, which makes the process more efficient. Then, we performed a detection process on each mutation with a probability of being detected of 90/113. At last, we recorded the observed number of mutations in each observed CHD gene of each mouse. The process is shown in Figure 4.4.

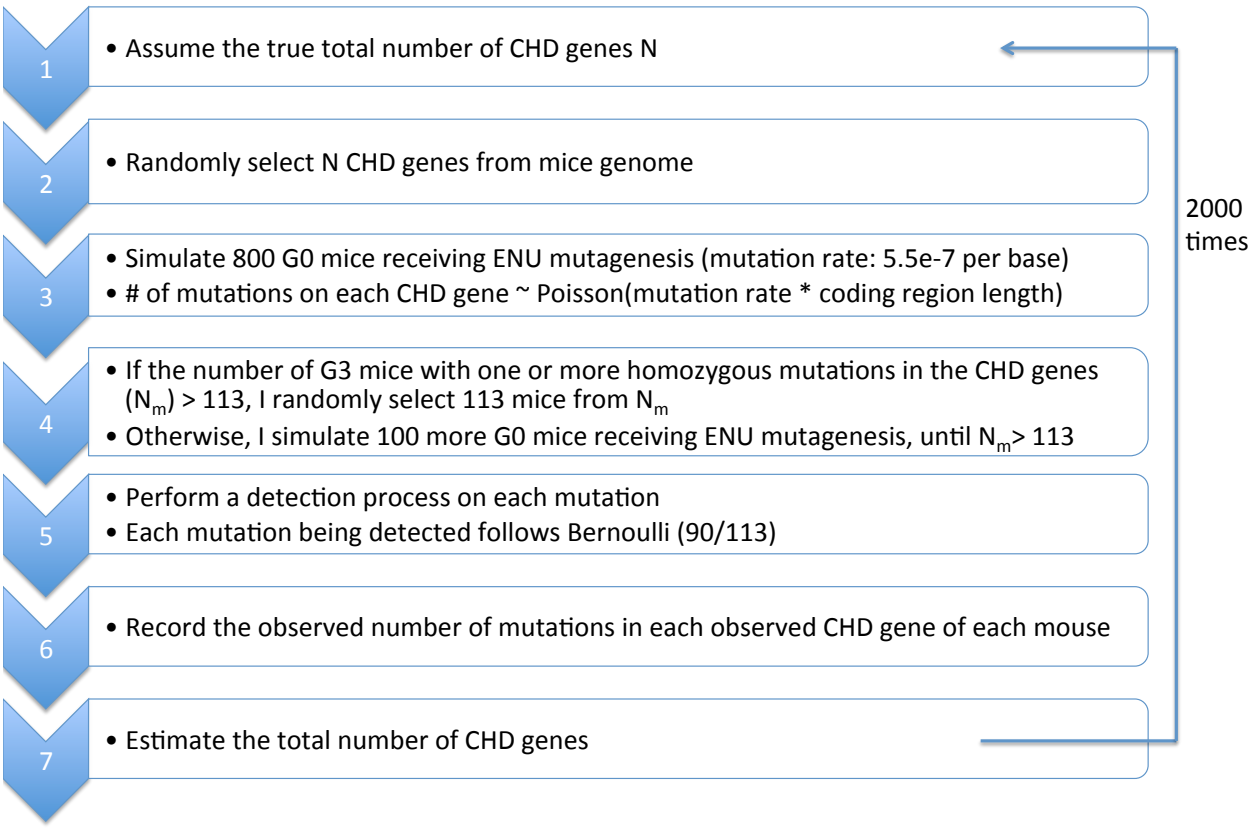


Figure 4.4: **Simulation process of the recessive forward genetic screening study based on the information from a real study.**

Table 4.1: **Mice family structure information.**

a.

The number of G2 mice that G1 mice mate with	Frequency
1	5
2	11
3	19
4	32
5	46
6	23
10	1

b.

The number of G3 mice generated	Frequency
2	66
3	68
4	74
5	46
6	47
7	40
8	35
9	26
10	14
11	10
12	9
13	7
14	5
15	5
16	1
18	2
19	1
20	1
21	3
25	1

The mice family structure information is provided by Li et al. [2015]. **a.** The contingency table shows the number of G2 mice that G1 mice mate with. **b.** The contingency table shows the number of G3 mice generated by their G1 father and G2 mother.

4.2.3 Total Number of Disease Genes Estimation evaluation

We set the true total number of disease genes as 400, 600, and 800, respectively. For each total number of disease genes, we simulated the RFGSS process for 2000 times and estimated the total number of disease genes in each simulation. We evaluated the accuracy and precision of the proposed estimator by taking the average and the standard deviation of the 2000 estimates. We also evaluated the 90% quantile interval by calculating its coverage rate.

4.2.4 Methods comparison

With the data being observed in each simulation process, we compared our proposed estimator with the nonparametric estimators first. We compared our proposed estimator with two estimators proposed by Chao [1984, 1992], a jackknife estimator proposed by Burnham and Overton [1978; 1979], an estimator derived from Poisson-Gamma model proposed by Chao and Bunge [2002], a nonparametric maximum likelihood estimator proposed by Norris and Pollock [1996; 1998], a penalized nonparametric maximum likelihood estimator proposed by Wang and Lindsay [2005; 2008]. We ran the above methods using SPECIES R package [Wang et al., 2011]. These methods are also compared with the "$\gamma = 1$ unseen species" method applied in Sanders et al. [2012] and Zaidi et al. [2013]. We used the estimated total numbers from these methods to compare with the truth we set at the beginning of the simulation. Then, we applied the parametric methods that are widely used proposed by Pollock and Larkin [2004] to the data being observed in each simulation process and compared the results to the proposed method and the truth. The parametric methods generate the maximum likelihood estimates assuming the mutations on the observed disease genes follow zero-truncated negative binomial distribution, zero-truncated Poisson distribution with a single mutation rate, zero-truncated mixture Poisson distribution with two different mutation rates, and zero-truncated mixture Poisson distribution with three different mutation rates, respectively.

4.3 RESULTS

We introduced the comparison of the eight nonparametric methods described in "Methods comparison" section. In Table 4.2, Table 4.3, Table 4.4, and Figure 4.5, "proposed" represents the proposed simulation-based "unseen species" method; "γ=1" represents the "$\gamma = 1$ unseen species" method applied in Sanders et al. [2012] and Zaidi et al. [2013]; "CL1984" and "CL1992" represent the two methods provided by Chao [1984, 1992], re-

spectively; "ChaoBunge" represents Chao and Bunge's Poisson-Gamma method [Chao and Bunge, 2002]; "jackknife" represents Burnham and Overton's jackknife method [Burnham and Overton, 1978, 1979]; "pnpmle" represents Norris and Pollock's nonparametric maximum likelihood method [Norris and Pollock, 1996, 1998]; and "unpmle" represents Wang and Lindsay's penalized nonparametric maximum likelihood method [Wang and Lindsay, 2005, 2008]. Table 4.2, Table 4.3 and Figure 4.5 shows that the "$\gamma = 1$ unseen species" method provide the least biased estimate, whose mean and median values are closest to the truth, but slightly bigger than the truth. The proposed simulation-based "unseen species" method over estimates the total number of disease genes. Jackknife method and the Chao and Bunge's Poisson-Gamma method have the most biased estimates compared to the others, with the mean and median values farthest from the truth. Table 4.4 and Figure 4.5 show that Jackknife method provides the estimates with the smallest standard deviation. The proposed simulation-based "unseen species" method, "$\gamma = 1$ unseen species" method, two Chao's method, and Norris and Pollock's nonparametric maximum likelihood method have similar standard deviations, bigger than that of jackknife method. Chao and Bunge's nonparametric maximum likelihood method and Wang and Lindsay's penalized nonparametric maximum likelihood method have much bigger standard deviation than the others.

We also compared the proposed method and the "$\gamma = 1$ unseen species" method to the parametric methods. Assuming that the mutations on the observed disease genes follow a zero-truncated negative binomial distribution generates huge and unrealistic total number of disease genes estimates. So we did not show the result of this method in the dissertation. In Table 4.5, Table 4.6, Table 4.7, and Figure 4.6, "proposed" and "$\gamma = 1$" have the same meaning as above. "Poisson" represents the parametric method assuming that the mutations on the observed disease genes follow a zero-truncated Poisson distribution with a single mutation rate. "C2" represents the parametric method assuming that the mutations on the observed disease genes follow a zero-truncated mixture Poisson distribution with two different mutation rates. "C3" represents the parametric method assuming that the mutations on the observed disease genes follow a zero-truncated mixture Poisson distribution with three different mutation rates. Table 4.5, Table 4.6 and Figure 4.6 show that the parametric methods underestimate the total number of disease genes a lot. Table 4.7 shows that the

standard deviations of the parametric methods estimates are also smaller than the proposed method and the "$\gamma = 1$ unseen species" method.

We evaluated the 90% quantile interval of the proposed simulation based "unseen species" estimates by calculating its coverage rate. When the true total numbers of disease genes are 400, 600, and 800, the coverage rates are 0.17, 0.17, and 0.14, respectively.

All the methods, including the parametric ones and the nonparametric ones, generate some extreme estimates. An "extreme estimate" is defined as one which is either 1.5 times interquartile range (IQR) or more above the third quartile or 1.5 times IQR or more below the first quartile, where the IQR is the difference between the first quartile and the third quartile. Below, we only explore the extreme estimates for the proposed method and the "$\gamma = 1$ unseen species" method. All corresponding estimates generated by the two methods are based on the same observed data set. When the true total number of disease genes is 400, 600, and 800, the proposed method generates 155, 169, and 159 extreme estimates, respectively; while the "$\gamma = 1$ unseen species" method generates 72, 78, and 105 extreme estimates, respectively. If the true total number of disease genes is 400, 600, and 800, when "$\gamma = 1$ unseen species" estimates are extreme estimates, based on the same observed data, 98.6%, 100% and 100% of the estimates generated by the proposed methods are also extreme estimates, respectively; when the estimates by the proposed method are extreme estimates, based on the same observed data, only 45.8%, 46.1% and 66.0% of the "$\gamma = 1$ unseen species" estimates are also extreme estimates, respectively (Table 4.8).

Table 4.2: **The mean values of 2000 estimates by the eight nonparametric methods.**

	proposed	γ=1	CL1992	jackknife	CL1984	ChaoBunge	pnpmle	unpmle
400	491.182	444.675	348.265	215.017	284.683	80.701	310.565	7454.462
600	744.573	662.996	522.016	220.450	410.834	Inf	477.01	20328.708
800	1014.960	904.577	700.122	220.534	551.186	Inf	731.008	41487.343

Table 4.3: **The median values of 2000 estimates by the eight nonparametric methods.**

	proposed	γ=1	CL1992	jackknife	CL1984	ChaoBunge	pnpmle	unpmle
400	458.300	423.621	301.000	207.000	265.000	177.000	280.000	270.000
600	671.500	606.300	411.000	213.000	366.000	212.000	378.500	361.000
800	914.400	810.188	546.000	215.000	477.000	225.000	496.000	470.000

Table 4.4: **The standard deviations of 2000 estimates by the eight nonparametric methods.**

	proposed	γ=1	CL1992	jackknife	CL1984	ChaoBunge	pnpmle	unpmle
400	158.400	129.102	179.827	45.023	100.971	2931.769	303.147	44304.815
600	279.483	249.557	473.659	52.312	194.441	Inf	917.395	194734.285
800	417.007	430.025	660.495	60.555	317.249	Inf	2189.892	418812.350

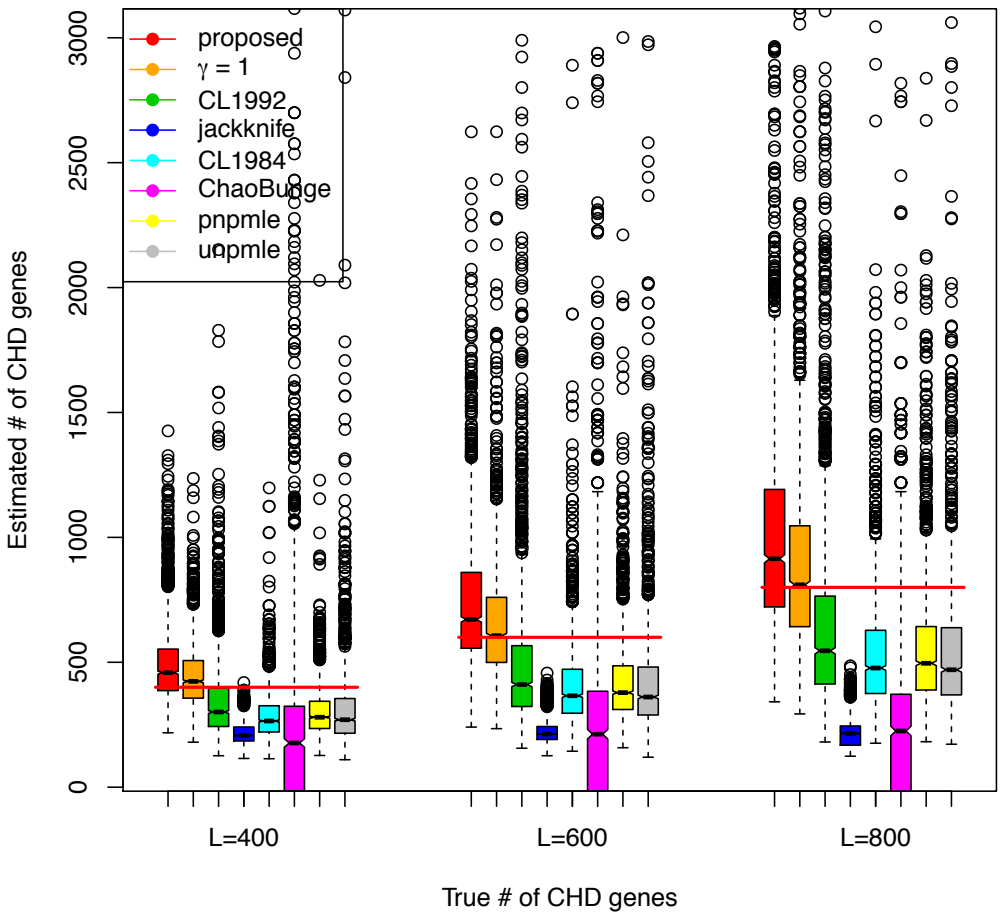


Figure 4.5: **Comparisons of the eight nonparametric estimates: the horizontal line in each box plot shows the median value. The red horizontal line shows the true total number of disease genes set at the beginning of the simulation.**

Table 4.5: **The mean values of 2000 estimates of the proposed method, "γ=1 unseen species" method, and the three parametric methods.**

	proposed	γ=1	Poisson	C2	C3
400	491.182	444.675	231.366	261.739	274.669
600	744.573	662.996	339.389	369.110	384.450
800	1014.960	904.577	459.037	486.595	503.019

Table 4.6: **The median values of 2000 estimates of the proposed method, "γ=1 unseen species" method, and the three parametric methods.**

	proposed	γ=1	Poisson	C2	C3
400	458.300	423.621	218.926	252.927	265.576
600	671.500	606.300	314.530	343.880	359.265
800	914.400	810.188	410.035	442.095	458.440

Table 4.7: **The standard deviations of 2000 estimates of the proposed method, "γ=1 unseen species" method, and the three parametric methods.**

	proposed	γ=1	Poisson	C2	C3
400	158.400	129.102	65.901	67.067	72.926
600	279.483	249.557	124.989	124.123	126.862
800	417.007	430.025	217.178	212.951	210.941

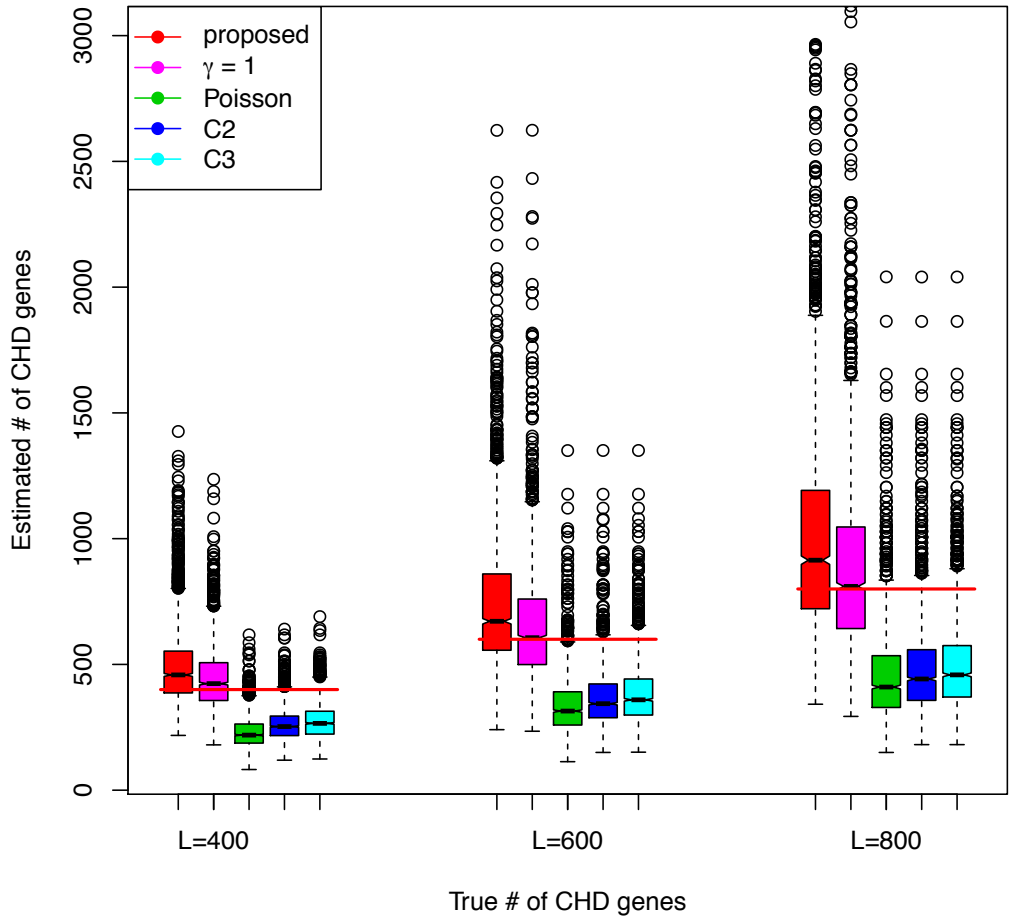


Figure 4.6: **Comparisons of the proposed method, "γ=1 unseen species" method, and the three parametric methods: the horizontal line in each box plot shows the median value. The red horizontal line shows the true total number of disease genes set at the beginning of the simulation.**

4.4 DISCUSSION

For an RGFSS, a desirable method would have the following properties: 1) it generates the estimates with mean and median close to the truth (unbiasedness); 2) it generates estimates with a small standard deviation. Among the 8 nonparametric methods, the proposed method

performs better than all the others, except the "$\gamma = 1$ unseen species" method. The proposed method overestimates the total number of disease genes but it provides estimates with relatively small standard deviation. The "$\gamma = 1$ unseen species" method slightly overestimates the total number of disease genes, but it generates the estimates that are most unbiased and have relatively small standard deviation. The two Chao's method [Chao and Lee, 1992; Chao, 1984], the jackknife estimator proposed by Burnham and Overton [1978; 1979], the estimator derived from Poisson-Gamma model proposed by Chao and Bunge [2002], the nonparametric maximum likelihood estimator proposed by Norris and Pollock [1996; 1998], and the penalized nonparametric maximum likelihood estimator proposed by Wang and Lindsay [2005; 2008] provide unrealistic results. Therefore, they are not appropriate to be applied to the RFGSS.

Although in our mice model based simulated RFGSS, the "$\gamma = 1$ unseen species" method performs better than the proposed method, the proposed method is still a promising method. The "$\gamma = 1$ unseen species" method may perform better because the true γ is close to 1, where γ depends on the length of the coding regions of the disease genes. If the RFGSS is conducted on other animals, the true γ could be much bigger or smaller than 1, and then "$\gamma = 1$ unseen species" method may no longer work better than the proposed method. Thus, compared to the "$\gamma = 1$ unseen species" method, the proposed method is more flexible, because it does not assume a priori that gamma equals 1, and it is appropriate to apply to other animal based RFGSS.

It is not appropriate to use the parametric methods to estimate the total number of disease genes in the RFGSS. As Table 4.6 and Figure 4.6 in the "results" section shown, the parametric methods underestimate the total number of disease genes.

The 90% coverage rates of the proposed method's quantile interval are as low as 0.17, 0.17 and 0.14, when the true total numbers of disease genes are 400, 600, and 800, respectively. The main reason for the low coverage rate is that the proposed method overestimates the total numbers of disease genes. Thus, the true total number of disease genes are less likely to be included within the quantile intervals.

From Figure 4.5, all the eight nonparametric methods have some extreme estimates, and from Figure 4.6, all the three parametric methods have some extreme estimates. The

distribution of the observed mutations on the observed disease genes may be one possible cause of the extreme estimates. Here, we discuss the extreme estimates generated from the "$\gamma = 1$ unseen species" method and the proposed simulation-based "unseen species" method. Using the same sets of the observed disease mutations, when the "$\gamma = 1$ unseen species" estimates are extreme estimates, almost all the proposed simulation-based "unseen species" estimates are extreme estimates; but when the proposed simulation-based "unseen species" estimates are extreme estimates, only half of the "$\gamma = 1$ unseen species" estimates are extreme estimates (Table 4.8). The reason may be that the extreme estimates generated from the "$\gamma = 1$ unseen species" method are due to the distribution of the observed disease mutations on the observed disease genes, but the the extreme estimates generated from the proposed method are due to both the distribution of the observed disease mutations on the observed disease genes, and the simulated unobserved mutations on the sampled unobserved genes, which generates the extreme γ's and leads to the extreme estimates. In addition, while the extreme estimates are a function of the observed data, they could be due to unmet or inappropriate assumptions of the different methods.

Assessing the patterns of the distributions of the observed mutations on observed genes that generate the extreme estimates is necessary. Since all the methods generate some extreme estimates, when people conduct a RFGSS and obtain only one distribution of the observed mutations on observed genes, it is unclear whether the subsequent single estimate of the total number of disease genes is an extreme estimate or not. In RFGSS, erroneously large estimates may lead to an inappropriate screening strategy, thus consuming a large amount of follow-up effort and money unnecessarily. If people aim to find all the undetected disease genes, but the estimated total number of them is much more than the truth, expensive follow-up experiments may be conducted, screening more animals than necessary. Assessing the patterns of the distributions of the observed mutations on observed genes that generate the extreme estimates may provide the investigator with information regarding whether they have a spuriously large estimate, and thus, may guide the usage of the methods to estimate the total number of disease genes.

Our research provides practical guidance regarding estimating the total number of disease genes from a RFGSS data set. Given a RFGSS data set, based on what we learned in the

study, we would recommend using the "$\gamma = 1$ unseen species" method to estimate the total number of disease genes. But we would ask people to be cautious that the estimates could be extreme.

Table 4.8: **Comparisons of the extreme estimates generated by the proposed method and the "$\gamma = 1$ unseen species" method, when the true total number of disease genes is 400 (a), 600 (b), 800 (c), respectively.**

a.

Truth = 400		Proposed method		
		Extreme	Not extreme	Sum up
"$\gamma = 1$" method	Extreme	71	1	72
	Not extreme	84	1844	1928
	Sum up	155	1845	2000

b.

Truth = 600		Proposed method		
		Extreme	Not extreme	Sum up
"$\gamma = 1$" method	Extreme	78	0	78
	Not extreme	91	1831	1922
	Sum up	169	1831	2000

c.

Truth = 800		Proposed method		
		Extreme	Not extreme	Sum up
"$\gamma = 1$" method	Extreme	105	0	105
	Not extreme	54	1841	1895
	Sum up	159	1841	2000

5.0 FUTURE WORK

5.1 GENETIC RISK MODELS AND SCREENING STRATEGIES: INFLUENCE OF MODEL SIZE ON RISK ESTIMATES AND PRECISION

This study constructs better risk prediction models and provided a better screening strategy by taking the confidence interval of the predicted risk into account. The main focus of the study is not providing accurate risks, but to guide the screening decision relative to a fixed threshold. From the result of the coverage probabilities, the maxMRS-selected model has relatively low coverage probabilities (Figure 2.6), thus it may not be the best model to provide accurate risk estimates. In the future, we would like to further investigate how accurate risk estimates affect screening decisions, where the accurate risk estimates are those with true risks locating inside the confidence intervals of the estimates. Accordingly, inaccurate risk estimates are those with true risks located outside the confidence intervals. Since we know the true risks in the simulation studies but not in the real studies, this investigation would be conducted using simulated data sets.

There are eight possible relationships between the inaccurate risk estimates and the accurate risk estimates to the true risk and the screening threshold (Figure 5.1). If the true risk is bigger than the threshold, the inaccurate risk estimates can have their confidence intervals' upper bounds and the lower bounds both bigger than the true risk (Figure 5.1a situation 1, 2), both smaller than the threshold (Figure 5.1a situation 7, 8), both between the true risk and the threshold (Figure 5.1a situation 3, 4), or both smaller than the true risk but overlapped with the threshold (Figure 5.1a situation 5, 6). Accurate risk estimates can have their confidence interval upper bounds bigger than the true risk and lower bounds

between the true risk and the threshold (Figure 5.1a situation 1, 3, 5, 7), or upper bounds bigger than the true risk and lower bounds smaller than the threshold (Figure 5.1a situation 2, 4, 6, 8). Using the screening strategy {-T, 1], in Figure 5.1a situation 1 to 6, the same screening decisions are made with both accurate and inaccurate risk estimates. Both estimates correctly classified the individual to high risk category. In Figure 5.1a situation 7 and 8, inaccurate risk estimates incorrectly classified the individual and accurate risk estimates correctly classified the individual. Similarly, if the true risk is smaller than the threshold, the inaccurate risk estimates can have their confidence intervals' upper bounds and the lower bounds both bigger than the threshold (Figure 5.1b situation 7, 8), both smaller than the true risk (Figure 5.1b situation 1, 2), both between the true risk and the threshold (Figure 5.1b situation 3, 4), or both bigger than the true risk but overlapping the threshold (Figure 5.1b situation 5, 6). The accurate risk estimates can have their confidence interval lower bounds smaller than the true risk and upper bounds between the true risk and the threshold (Figure 5.1b situation 1, 3, 5, 7), or lower bounds smaller than the true risk and upper bound bigger than the threshold (Figure 5.1b situation 2, 4, 6, 8). Using the screening strategy {-T, 1], in Figure 5.1b situation 1, 3, 6, and 8, the same screening decisions are made with the accurate and the inaccurate risk estimates. In Figure 5.1b situation 2 and 4, inaccurate risk estimates correctly classified the individual and accurate risk estimates incorrectly classified the individual. In Figure 5.1b situation 5 and 7, inaccurate risk estimates incorrectly classified the individual and accurate risk estimates correctly classified the individual.

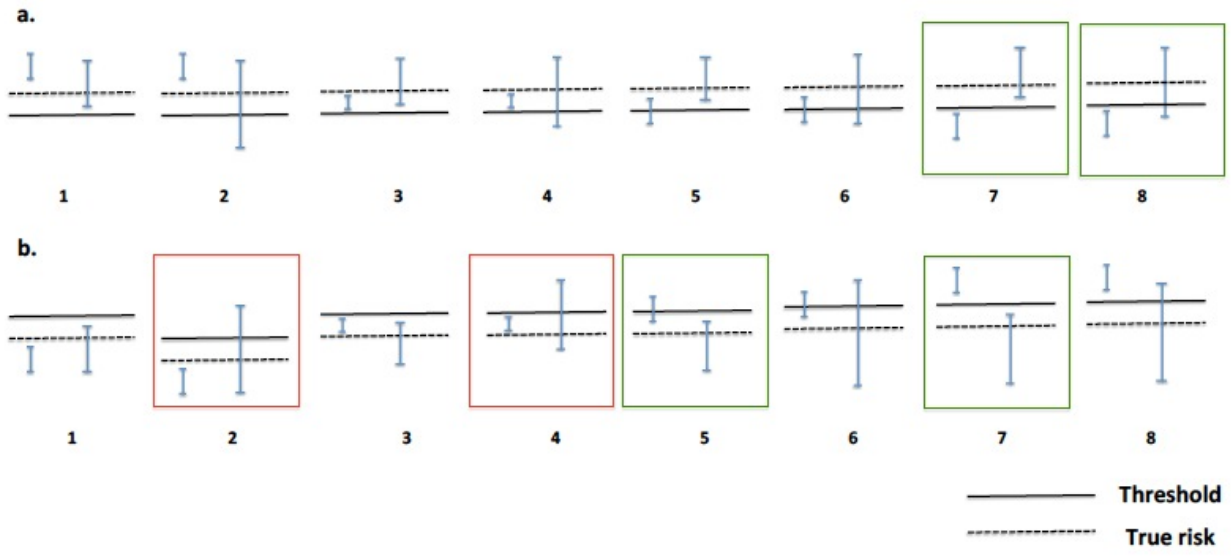


Figure 5.1: **Eight possible relationships for the inaccurate risk estimates and the accurate risk estimates to the true risks and the thresholds for cases (a) and controls (b), respectively.**

The green squares indicate that the inaccurate risk estimates incorrectly classified the individuals, while the accurate risk estimates correctly classified the individuals. The red squares indicate that the inaccurate risk estimates correctly classified the individuals, while the accurate risk estimates incorrectly classified the individuals. The classification is based on the screening strategy {-T, 1].

To explore how accurate risk estimates affect screening decisions, we would collect the individuals in the simulation data set who have inaccurate risk estimates under the maxMRS-selected models and have accurate risk estimates under the full model. We would measure how often the inaccurate risk estimates under the maxMRS-selected models and the accurate risk estimates under the full models provide the same correct/incorrect screening decisions. And how often the accurate risk estimates provide the correct/incorrect screening decisions, while the inaccurate risk estimates provide the incorrect/correct screening decision. Then, we would generate a 2×2 contingency table and conduct a chi-square test to assess the null hypothesis that the number of correct screening decisions are made on individuals having inaccurate estimates under the maxMRS-selected model are not significantly different from the number of correct screening decisions are made on individuals having accurate estimates

under the full model. Thus, we can investigate how the accurate risk estimates affect the screening decisions.

5.2 A BAYESIAN APPROACH FOR SNP EFFECT ESTIMATION AND GENETIC RISK PREDICTION FOR DICHOTOMOUS TRAITS

In the future, we would like to optimize the BayesRB program to accelerate its computation speed and improve it to use less memory. The current version of BayesRB program is not optimal. Currently, on a Red Hat Linux machine with 128 GB of RAM and a 2.1 GHz AMD Opteron Processor 6272, it takes around 3 days and 14 GB of memory to run 15,000 loops using data set with 3,000 individuals, where each was genotyped at around 300K SNPs. BayesR only takes 8 hours and 11.5 GB to process the same data set. There is still a lot of room for the improvement of the speed and the efficiency of BayesRB. We will optimize the program by considering the data storage and data reading of C++. For example, since in C++, high dimensional arrays are stored in one dimensional memory, accessing data in a sequential fashion as stored in physical memory, can speed up the program.

When applying BayesRB to the genome-wide simulated data sets and the real data sets, BayesRB only ran 15,000 and 13,000 loops, respectively, both with 5000 loops as warm-up. Some parameters do not mix well. In the future, after the speeding up of BayesRB, we would run BayesRB for more iterations to let all the parameters mix well.

What's more, we would like to improve the BayesRB program to enable it to handle missing phenotype data. The current version of BayesRB program does not allow either missing phenotype nor genotype data. In the future, for individuals with missing phenotypes, BayesRB will be able to detect them and delete them from the input matrix. For the missing genotypes, we would recommend using high quality imputation software, such as BEAGLE [Browning and Browning, 2016] or IMPUTE2 [Howie et al., 2009], to fill in any missing genotypes before running BayesRB.

We would systematically search for all the SNPs that are affected by the batch effects. This could be complemented by some automated searching of the intensity cluster plots for

those that did not cluster nicely into the expected three clusters. GWASTools [Gogarten et al., 2012] from Bioconductor and gPCA [Reese et al., 2013] are possible methods to do so. In Aim2, BayesRB discovered that rs11887827 in the CD data set and rs12050604 in the BD data set are affected by batch effects. For both rs11887827 and rs12050604, different decisions were made in controls and cases when calling the genotypes in cluster 2 (Figure 3.28). For both SNPs, controls in cluster 2 are treated as having 1/0 genotypes, but cases in cluster 2 are treated as having 1/1 genotypes. We would remove rs11887827 from CD data set and rs12050604 from BD data set, as well as all the other SNPs affected by the batch effects, and then measure the bayesRB performance again to allow more accurate performance comparisons.

5.3 A SIMULATION BASED "UNSEEN SPECIES" METHOD TO ESITMATE THE TOTAL NUMBER OF DISEASE GENES IN A RECESSIVE FORWARD GENETIC SCREENING STUDY (RFGSS)

In this study, both the parametric methods and the nonparametric methods generate extreme estimates, thus, the distribution of the estimates is skewed. The extreme estimates of all the methods, except the proposed "simulation-based unseen species" method, may due to certain distributions of the number of observed mutations on the observed disease genes. The extreme estimates of the proposed method may be due to both the distributions of the number of observed mutations on the observed disease genes and the distributions of the number of simulated unobserved mutations on the sampled unobserved disease genes. We could adjust the estimates by transforming the skewed distributions to normal distributions. We would assess whether the transformation method works by separating the data sets to 80/20 training and testing data sets. We would seek the best transformation method using the training data sets and test the transformation performance using the testing data sets.

In the future, we would like to assess what patterns in the distributions lead to the extreme estimates for each method. We would first record the distributions which generate extreme estimates using each method in the 2000 replicates of the simulation study. Then,

we would produce some descriptive statistics. We would also visualize the distributions that generate the extreme estimates using bar plots to assess if there is any pattern in those distributions. If there is a pattern in those distributions, we would investigate what leads to the pattern, thus leads to the extreme estimates. Our hypothesis is that, in the simulation, when genes which are extremely long are selected as the disease genes, after the RFGSS, the observations tend to generate extreme estimates. We would conduct chi-square tests to assess this hypothesis. With an idea of what patterns in the observed distributions lead to the extreme estimates and what lead to the patterns, we may be able to improve the statistics and the program may be able to output diagnostic information after conducting RFGSS. Also investigators may be able to decide whether each method is good to use for their observed distributions.

In addition, our simulation is based on the mouse genome, which may have a true γ close to 1, thus, leading to the result that "$\gamma = 1$" method performs better than the proposed method. In the future, we would like to assess whether the true γ is different from 1. Our null hypothesis is that the true γ's in the 2000 simulation replicates are not significantly different from 1. We would record the true γ's in the 2000 replicates and conduct a t test to assess the hypothesis. Furthermore, we would like to compare how "$\gamma = 1$" method and the proposed methods perform when the true γ is different from 1. We would select an organism where the true $\gamma \neq 1$. Then, using the same RFGSS simulation process as that in Aim 3, we could compare the accuracy and the precision of the total number of disease genes estimation by the two methods.

APPENDIX A

DETAILED MCMC STEPS IN THE BAYESRB ALGORITHM

Full conditional distribution for the general mean μ (MCMC step 2)

The logistic regression model can be written as a linear form with the latent variable Z_i.

$Z_i = \mu + \sum_{j=1}^{p} X_{ij}\beta_j + \epsilon_i$, with $\epsilon_i \overset{iid}{\sim} N(0, \lambda_i)$.

So, $\mu = Z_i - \sum_{j=1}^{p} X_{ij}\beta_j - \epsilon_i$

Then, $n\mu = \sum_{i=1}^{n}(Z_i - \sum_{j=1}^{p} X_{ij}\beta_j) - \sum_{i=1}^{n} \epsilon_i$, where $\sum_{i=1}^{n} \epsilon_i \sim N(0, \sum_{i=1}^{n} \lambda_i)$.

Therefore, the conditional posterior distribution of the general mean μ is

$\mu|. \sim N(n^{-1}\sum_{i=1}^{n}(Z_i - \sum_{j=1}^{p} X_{ij}\beta_j), \frac{\sum_{i=1}^{n}\lambda_i}{n^2})$

Proof of the full conditional posterior distribution of b_j and β_j (MCMC step 3)

For SNP j, we update β_j right after b_j, and then we update them for SNP $j+1$, which means we update β_j and b_j jointly.

$P(b_j, \beta_j|.)$

$= P(b_j, \beta_j|\boldsymbol{Z}, X, \sigma_g^2, \boldsymbol{\lambda}, \mu, \boldsymbol{\beta_{-j}}, \boldsymbol{b_{-j}}, \boldsymbol{\pi})$

$\propto P(\beta_j|b_j, \boldsymbol{Z}, X, \sigma_g^2, \boldsymbol{\lambda}, \mu, \boldsymbol{\beta_{-j}}, \boldsymbol{b_{-j}}, \boldsymbol{\pi}) P(b_j|\boldsymbol{Z}, X, \sigma_g^2, \boldsymbol{\lambda}, \mu, \boldsymbol{\beta_{-j}}, \boldsymbol{b_{-j}}, \boldsymbol{\pi})$,

where $\boldsymbol{b_{-j}}$ denotes the vector of categories that all the SNPs expect SNP j belong to. $\boldsymbol{\beta_{-j}}$ denotes the vector of effects of all the SNPs expect SNP j.

1) First b_j is updated by calculating $P(b_j|\boldsymbol{Z}, X, \sigma_g^2, \boldsymbol{\lambda}, \mu, \boldsymbol{\beta_{-j}}, \boldsymbol{b_{-j}}, \boldsymbol{\pi})$:

$P(b_j = k|X, \boldsymbol{Z}, \sigma_g^2, \boldsymbol{\lambda}, \mu, \boldsymbol{\beta_{-j}}, \boldsymbol{\pi})$

$= \frac{P(b_j=k, \boldsymbol{Z}|X, \sigma_g^2, \boldsymbol{\lambda}, \mu, \boldsymbol{\beta_{-j}}, \boldsymbol{\pi})}{P(\boldsymbol{Z}|X, \sigma_g^2, \boldsymbol{\lambda}, \mu, \boldsymbol{\beta_{-j}}, \boldsymbol{\pi})}$

$= \frac{P(b_j=k,\boldsymbol{Z}|X,\sigma_g^2,\boldsymbol{\lambda},\mu,\boldsymbol{\beta_{-j}},\boldsymbol{\pi})}{\sum_{l=1}^{4} P(b_j=l,\boldsymbol{Z}|X,\sigma_g^2,\boldsymbol{\lambda},\mu,\boldsymbol{\beta_{-j}},\boldsymbol{\pi})}$,

where k is the category SNP j is assigned to. $k = (1, 2, 3, 4)$.

Set $L_{jk} = P(b_j = k, \boldsymbol{Z}|X, \sigma_g^2, \boldsymbol{\lambda}, \mu, \boldsymbol{\beta_{-j}}, \boldsymbol{\pi})$,

Then, $P(b_j = k|X, \boldsymbol{Z}, \sigma_g^2, \boldsymbol{\lambda}, \mu, \boldsymbol{\beta_{-j}}, \boldsymbol{\pi})$

$= \frac{L_{jk}}{\sum_{l=1}^{k} L_{jl}}$

$= \frac{1}{\sum_{l=1}^{4} exp(logL_{jl} - logL_{jk})}$.

L_{jk} is calculated below.

$L_{jk} = P(b_j = k, \boldsymbol{Z}|X, \sigma_g^2, \boldsymbol{\lambda}, \mu, \boldsymbol{\beta_{-j}}, \boldsymbol{\pi})$

$= P(\boldsymbol{Z}|X, b_j = k, \sigma_g^2, \boldsymbol{\lambda}, \mu, \boldsymbol{\beta_{-j}}, \boldsymbol{\pi}) P(b_j = k|X, \sigma_g^2, \boldsymbol{\lambda}, \mu, \boldsymbol{\beta_{-j}}, \boldsymbol{\pi})$

$= P(\boldsymbol{Z}|X, b_j = k, \sigma_g^2, \boldsymbol{\lambda}, \mu, \boldsymbol{\beta_{-j}}, \boldsymbol{\pi}) P(b_j = k|\boldsymbol{\pi})$

$= \int_{\beta_j} P(\boldsymbol{Z}|X, b_j = k, \sigma_g^2, \boldsymbol{\lambda}, \mu, \boldsymbol{\beta_{-j}}, \boldsymbol{\pi}, \beta_j) P(\beta_j|X, b_j = k, \sigma_g^2, \boldsymbol{\lambda}, \mu, \boldsymbol{\beta_{-j}}, \boldsymbol{\pi})$

$\times P(b_j = k|\boldsymbol{\pi}) d\beta_j$

For individual i,

$Z_i = \mu + X_{ij}\beta_j + \sum_{l \neq j} X_{il}\beta_l + \epsilon_i$, where $\epsilon_i \overset{iid}{\sim} N(0, \lambda_i)$.

$Z_i \overset{iid}{\sim} N(\mu + X_{ij}\beta_j + \sum_{l \neq j} X_{il}\beta_l, \lambda_i)$.

Therefore, $P(\boldsymbol{Z}|X, b_j = k, \sigma_g^2, \boldsymbol{\lambda}, \mu, \boldsymbol{\beta_{-j}}, \boldsymbol{\pi}, \beta_j)$

$= \prod_{i=1}^{n} \left[\frac{1}{\sqrt{2\pi(\lambda_i)}} exp\{-\frac{1}{2\lambda_i}(\tilde{Z}_{ij} - X_{ij}\beta_j)^2\} \right]$,

where $\tilde{Z}_{ij} = Z_i - \mu - \sum_{l \neq j} X_{il}\beta_l$.

$\boldsymbol{C}$ is the vector (0, 10^{-4}, 10^{-3}, 10^{-2}). Then,

$$C_{b_j} \sim \begin{cases} 0 & b_j = 1 \\ 10^{-4} & b_j = 2 \\ 10^{-3} & b_j = 3 \\ 10^{-2} & b_j = 4 \end{cases}$$

.

If $b_j \neq 1$,

$P(\beta_j|X, b_j = k, \sigma_g^2, \boldsymbol{\lambda}, \mu, \boldsymbol{\beta_{-j}}, \boldsymbol{\pi}) = P(\beta_j|b_j = k, \sigma_g^2) = \frac{1}{\sqrt{2\pi C_{b_j}\sigma_g^2}} exp(-\frac{1}{2C_{b_j}\sigma_g^2}\beta_j^2)$,

which is the prior of β_j given β_j is assigned to the category b_j.

Set $V_j = \sum_{i=1}^{n} \frac{X_{ij}^2}{\lambda_i} + \frac{1}{C_{b_j}\sigma_g^2}$.

Then, $P(\boldsymbol{Z}|X, b_j = k, \sigma_g^2, \boldsymbol{\lambda}, \mu, \boldsymbol{\beta_{-j}}, \boldsymbol{\pi}, \beta_j) P(\beta_j|X, b_j = k, \sigma_g^2, \boldsymbol{\lambda}, \mu, \boldsymbol{\beta_{-j}}, \boldsymbol{\pi})$

$$\propto \left[(\prod_{i=1}^{n}\lambda_i^{-\frac{1}{2}})(C_{b_j}\sigma_g^2)^{-\frac{1}{2}}\right] exp\left\{-\frac{1}{2}\left[\sum_{i=1}^{n}\frac{\tilde{Z_{ij}}^2-2\tilde{Z_{ij}}X_{ij}\beta_j+X_{ij}^2\beta_j^2}{\lambda_i}+\frac{\beta_j^2}{C_{b_j}\sigma_g^2}\right]\right\}$$

$$= \left[(\prod_{i=1}^{n}\lambda_i^{-\frac{1}{2}})(C_{b_j}\sigma_g^2)^{-\frac{1}{2}}\right] exp\left\{-\frac{1}{2}\left[(\sum_{i=1}^{n}\frac{X_{ij}^2}{\lambda_i}+\frac{1}{C_{b_j}\sigma_g^2})\beta_j^2-2\sum_{i=1}^{n}\frac{\tilde{Z_{ij}}X_{ij}\beta_j}{\lambda_i}+\sum_{i=1}^{n}\frac{\tilde{Z_{ij}}^2}{\lambda_i}\right]\right\}$$

$$= \left[(\prod_{i=1}^{n}\lambda_i^{-\frac{1}{2}})(C_{b_j}\sigma_g^2)^{-\frac{1}{2}}\right] exp\left\{-\frac{1}{2\frac{1}{V_j}}\left[\beta_j^2-2\frac{\sum_{i=1}^{n}\frac{\tilde{Z_{ij}}X_{ij}\beta_j}{\lambda_i}}{V_j}+\frac{\sum_{i=1}^{n}\frac{\tilde{Z_{ij}}^2}{\lambda_i}}{V_j}\right]\right\}$$

$$= \left[(\prod_{i=1}^{n}\lambda_i^{-\frac{1}{2}})(C_{b_j}\sigma_g^2)^{-\frac{1}{2}}\right] exp\left\{-\frac{1}{2\frac{1}{V_j}}\left[\beta_j^2-2\frac{\sum_{i=1}^{n}\frac{\tilde{Z_{ij}}X_{ij}\beta_j}{\lambda_i}}{V_j}+(\frac{\sum_{i=1}^{n}\frac{\tilde{Z_{ij}}X_{ij}}{\lambda_i}}{V_j})^2\right]\right\}$$

$$\times exp\left\{-\frac{1}{2\frac{1}{V_j}}\left[-(\frac{\sum_{i=1}^{n}\frac{\tilde{Z_{ij}}X_{ij}}{\lambda_i}}{V_j})^2+\frac{\sum_{i=1}^{n}\frac{\tilde{Z_{ij}}^2}{\lambda_i}}{V_j}\right]\right\}$$

$$= \left[(\prod_{i=1}^{n}\lambda_i^{-\frac{1}{2}})(C_{b_j}\sigma_g^2)^{-\frac{1}{2}}\right] exp\left\{-\frac{1}{2\frac{1}{V_j}}\left[\beta_j-\frac{\sum_{i=1}^{n}\frac{\tilde{Z_{ij}}X_{ij}}{\lambda_i}}{V_j}\right]^2\right\}$$

$$\times exp\left\{-\frac{1}{2\frac{1}{V_j}}\left[-(\frac{\sum_{i=1}^{n}\frac{\tilde{Z_{ij}}X_{ij}}{\lambda_i}}{V_j})^2+\frac{\sum_{i=1}^{n}\frac{\tilde{Z_{ij}}^2}{\lambda_i}}{V_j}\right]\right\}.$$

So, $\int_{\beta_j} P(\boldsymbol{Z}|X,b_j=k,\sigma_g^2,\boldsymbol{\lambda},\mu,\boldsymbol{\beta_{-j}},\boldsymbol{\pi},\beta_j)P(\beta_j|X,b_j=k,\sigma_g^2,\boldsymbol{\lambda},\mu,\boldsymbol{\beta_{-j}},\boldsymbol{\pi})d\beta_j$

$$= \int_{\beta_j}\frac{1}{\sqrt{2\pi\frac{1}{V_j}}}exp\left\{-\frac{1}{2\frac{1}{V_j}}\left[\beta_j-\frac{\sum_{i=1}^{n}\frac{\tilde{Z_{ij}}X_{ij}}{\lambda_i}}{V_j}\right]^2\right\}d\beta_j$$

$$\times\sqrt{2\pi\frac{1}{V_j}}\left[(\prod_{i=1}^{n}\lambda_i^{-\frac{1}{2}})(C_{b_j}\sigma_g^2)^{-\frac{1}{2}}\right] exp\left\{-\frac{1}{2\frac{1}{V_j}}\left[-(\frac{\sum_{i=1}^{n}\frac{\tilde{Z_{ij}}X_{ij}}{\lambda_i}}{V_j})^2+\frac{\sum_{i=1}^{n}\frac{\tilde{Z_{ij}}^2}{\lambda_i}}{V_j}\right]\right\}$$

$$= \sqrt{2\pi\frac{1}{V_j}}\left[(\prod_{i=1}^{n}\lambda_i^{-\frac{1}{2}})(C_{b_j}\sigma_g^2)^{-\frac{1}{2}}\right] exp\left\{-\frac{1}{2\frac{1}{V_j}}\left[-(\frac{\sum_{i=1}^{n}\frac{\tilde{Z_{ij}}X_{ij}}{\lambda_i}}{V_j})^2+\frac{\sum_{i=1}^{n}\frac{\tilde{Z_{ij}}^2}{\lambda_i}}{V_j}\right]\right\}$$

If $b_j \neq 1$,

$$logL_{jk} = -\frac{1}{2}\left\{log(\prod_{i=1}^{n}\lambda_i)+log(\sum_{i=1}^{n}\frac{X_{ij}^2}{\lambda_i}C_{b_j}\sigma_g^2+1)\right\}-\frac{1}{2}\left[-\frac{(\sum_{i=1}^{n}\frac{\tilde{Z_{ij}}X_{ij}}{\lambda_i})^2}{V_j}+\sum_{i=1}^{n}\frac{\tilde{Z_{ij}}^2}{\lambda_i}\right]+$$

$log(\pi_k)$.

If $b_j = 1$,

$$logL_{j1} = -\frac{1}{2}\left\{log(\prod_{i=1}^{n}\lambda_i)\right\}-\frac{1}{2}\left[\sum_{i=1}^{n}\frac{\tilde{Z_{ij}}^2}{\lambda_i}\right]+log(\pi_1).$$

Set $T_k = P(b_j = k|\boldsymbol{Z},X,\sigma_g^2,\boldsymbol{\lambda},\mu,\boldsymbol{\beta_{-j}},\boldsymbol{b_{-j}},\boldsymbol{\pi})$

The SNP j is assigned to category k based on a value h sampled from a uniform distribution.

$$updated\ b_j = \begin{cases} 1 & if\ 0 < h \leq T_1 \\ 2 & if\ T_1 < h \leq T_1+T_2 \\ 3 & if\ T_1+T_2 < h \leq T_1+T_2+T_3 \\ 4 & if\ T_1+T_2+T_3 < h \leq 1. \end{cases}$$

2) Then, we update β_j by calculating $P(\beta_j|b_j,\boldsymbol{Z},X,\sigma_g^2,\boldsymbol{\lambda},\mu,\boldsymbol{\beta_{-j}},\boldsymbol{b_{-j}},\boldsymbol{\pi})$:

$P(\beta_j|b_j, \boldsymbol{Z}, X, \sigma_g^2, \boldsymbol{\lambda}, \mu, \boldsymbol{\beta_{-j}}, \boldsymbol{b_{-j}}, \boldsymbol{\pi})$

$= \frac{P(\boldsymbol{Z}|\beta_j, b_j, X, \sigma_g^2, \boldsymbol{\lambda}, \mu, \boldsymbol{\beta_{-j}}, \boldsymbol{b_{-j}}, \boldsymbol{\pi}) P(\beta_j|b_j, X, \sigma_g^2, \boldsymbol{\lambda}, \mu, \boldsymbol{\beta_{-j}}, \boldsymbol{b_{-j}}, \boldsymbol{\pi})}{\frac{P(b_j, \boldsymbol{Z}, X, \sigma_g^2, \boldsymbol{\lambda}, \mu, \boldsymbol{\beta_{-j}}, \boldsymbol{b_{-j}}, \boldsymbol{\pi})}{P(b_j, X, \sigma_g^2, \boldsymbol{\lambda}, \mu, \boldsymbol{\beta_{-j}}, \boldsymbol{b_{-j}}, \boldsymbol{\pi})}}$

$= \frac{P(\boldsymbol{Z}|\beta_j, b_j, X, \sigma_g^2, \boldsymbol{\lambda}, \mu, \boldsymbol{\beta_{-j}}, \boldsymbol{b_{-j}}, \boldsymbol{\pi}) P(\beta_j|b_j, X, \sigma_g^2, \boldsymbol{\lambda}, \mu, \boldsymbol{\beta_{-j}}, \boldsymbol{b_{-j}}, \boldsymbol{\pi})}{P(\boldsymbol{Z}|b_j, X, \sigma_g^2, \boldsymbol{\lambda}, \mu, \boldsymbol{\beta_{-j}}, \boldsymbol{b_{-j}}, \boldsymbol{\pi})}$

$\propto P(\boldsymbol{Z}|\beta_j, b_j, X, \sigma_g^2, \boldsymbol{\lambda}, \mu, \boldsymbol{\beta_{-j}}, \boldsymbol{b_{-j}}, \boldsymbol{\pi}) P(\beta_j|b_j, X, \sigma_g^2, \boldsymbol{\lambda}, \mu, \boldsymbol{\beta_{-j}}, \boldsymbol{b_{-j}}, \boldsymbol{\pi})$

$= P(\boldsymbol{Z}|\beta_j, X, \mu, \boldsymbol{\lambda}, \boldsymbol{\beta_{-j}}) P(\beta_j|\sigma_g^2, b_j)$

$\propto exp\{-\frac{1}{2}\sum_{i=1}^n \lambda_i^{-1}(\tilde{Z_{ij}} - X_{ij}\beta_j)^2\} exp\{-\frac{1}{2C_{b_j}\sigma_g^2}\beta_j^2\}$

$\propto exp\{-\frac{1}{2}(\sum_{i=1}^n \tilde{Z_{ij}}^2 \lambda_i^{-1} - 2\sum_{i=1}^n \tilde{Z_{ij}} X_{ij}\beta_j \lambda_i^{-1} + \sum_{i=1}^n X_{ij}^2 \beta_j^2 \lambda_i^{-1}) - \frac{1}{2C_{b_j}\sigma_g^2}\beta_j^2\}$

$\propto exp\{-\frac{1}{2\cdot\frac{1}{\sum_{i=1}^n X_{ij}^2 \lambda_i^{-1} + \frac{1}{C_{b_j}\sigma_g^2}}} \left[\beta_j^2 - \frac{2\sum_{i=1}^n \tilde{Z_{ij}} X_{ij} \lambda_i^{-1}}{\sum_{i=1}^n X_{ij}^2 \lambda_i^{-1} + \frac{1}{C_{b_j}\sigma_g^2}}\beta_j + Const\right]\}$, where $Const$ denotes a constant.

$\propto exp\{-\frac{1}{2\cdot\frac{1}{\sum_{i=1}^n X_{ij}^2 \lambda_i^{-1} + \frac{1}{C_{b_j}\sigma_g^2}}} \left[\beta_j - \frac{\sum_{i=1}^n \tilde{Z_{ij}} X_{ij} \lambda_i^{-1}}{\sum_{i=1}^n X_{ij}^2 \lambda_i^{-1} + \frac{1}{C_{b_j}\sigma_g^2}}\right]^2\}$, if $b_j \neq 1$.

Therefore, the conditional posterior distribution of the effect of SNP j, which belongs to category b_j, is

$$\beta_j|b_j, \boldsymbol{Z}, X, \sigma_g^2, \boldsymbol{\lambda}, \mu, \boldsymbol{\beta_{-j}}, \boldsymbol{b_{-j}}, \boldsymbol{\pi} \sim \begin{cases} \delta(\beta_j) & b_j = 1 \\ N(\frac{\sum_{i=1}^n \tilde{Z_{ij}} X_{ij} \lambda_i^{-1}}{\sum_{i=1}^n \lambda_i^{-1} X_{ij}^2 + \frac{1}{C_{b_j}\sigma_g^2}}, \frac{1}{\sum_{i=1}^n \lambda_i^{-1} X_{ij}^2 + \frac{1}{C_{b_j}\sigma_g^2}}) & b_j \neq 1 \end{cases},$$

where $\delta(\beta_j)$ denotes the dirac delta function with all probability mass at $\beta_j = 0$ if $b_j = 1$.

Full conditional distribution for the relative variance for each mixture component σ_g^2 (MCMC step 4)

We assumed the SNPs are independent. From the dependency diagram,

$P(\sigma_g^2|.)$

$= P(\sigma_g^2|\boldsymbol{b}, \boldsymbol{\beta})$

$= \frac{P(\boldsymbol{\beta}|\sigma_g^2, \boldsymbol{b}) P(\sigma_g^2|\boldsymbol{b})}{P(\boldsymbol{\beta}|\boldsymbol{b})}$

$\propto P(\boldsymbol{\beta}|\sigma_g^2, \boldsymbol{b}) P(\sigma_g^2)$

$\propto \prod_{k=1}^4 \left[\prod_{j:b_j=k} P(\beta_j|b_j = k, \sigma_g^2)\right] P(\sigma_g^2)$.

We use the uniform non-informative prior for σ_g^2. Then, $\sigma_g^2 \propto 1$.

$P(\sigma_g^2|.) \propto (\prod_{k=1}^4 (\sigma_g^2)^{-\frac{1}{2}m_k}) exp\left\{-\frac{1}{2}\sum_{k=1}^4 [C_k^{-1}\sigma_g^{-2}\sum_{j:b_j=k}\beta_j^2]\right\}$,

which is not a density function of any known distribution. Therefore, we used a Metropolis-Hasting sampling to update σ_g^2. The initial value of σ_g^2 ($\sigma_g^{2(0)}$) is sampled from N(0, U).

The steps are shown below:

1) Proposal function: $J_t(\sigma_g^{2(*)}|\sigma_g^{2(t)}) \sim N(\sigma_g^{2(t)}, \theta)$ truncated at 0.

$J_t(\sigma_g^{2(*)}|\sigma_g^{2(t)}) = \frac{1}{\sqrt{2\pi\theta}} exp[-\frac{1}{2\theta}(\sigma_g^{2(*)} - \sigma_g^{2(t)})^2]/\Phi(\frac{\sigma_g^{2(t)}}{\sqrt{\theta}})$

$J_t(\sigma_g^{2(t)}|\sigma_g^{2(*)}) = \frac{1}{\sqrt{2\pi\theta}} exp[-\frac{1}{2\theta}(\sigma_g^{2(*)} - \sigma_g^{2(t)})^2]/\Phi(\frac{\sigma_g^{2(*)}}{\sqrt{\theta}})$

2) $r = \frac{P(\sigma_g^{2(*)}|.)/J_t(\sigma_g^{2(*)}|\sigma_g^{2(t)})}{P(\sigma_g^{2t}|.)/J_t(\sigma_g^{2(t)}|\sigma_g^{2(*)})}$

$= \frac{P(\sigma_g^{2(*)}|.)\Phi(\frac{\sigma_g^{2(t)}}{\sqrt{\theta}})}{P(\sigma_g^{2t}|.)\Phi(\frac{\sigma_g^{2(*)}}{\sqrt{\theta}})}$

$= \frac{(\sigma_g^{2(*)})^{-\frac{1}{2}\sum_{k=2}^{4} m_k} exp\{-\frac{1}{\sigma_g^{2(*)}}[\frac{1}{2}\sum_{k=2}^{4}(C_k^{-1}\sum_{j:b_j=k}\beta_j^2)]\}\Phi(\frac{\sigma_g^{2(t)}}{\sqrt{\theta}})}{(\sigma_g^{2(t)})^{-\frac{1}{2}\sum_{k=2}^{4} m_k} exp\{-\frac{1}{\sigma_g^{2(t)}}[\frac{1}{2}\sum_{k=2}^{4}(C_k^{-1}\sum_{j:b_j=k}\beta_j^2)]\}\Phi(\frac{\sigma_g^{2(*)}}{\sqrt{\theta}})}$

3) Sample $v \sim unif(0,1)$

4) Update $\sigma_g^{2(t+1)} = \begin{cases} \sigma_g^{2(t)} & if\ v > r \\ \sigma_g^{2(*)} & if\ v \leq r \end{cases}$

In the above steps, U and θ are tuned manually. Using the unrealistic data set, $U = 200$ and $\theta = 1$.

APPENDIX B

PSEUDO CODE OF BAYESRB

%% Set initial values

$C[1:4] \leftarrow (0, 10^{-4}, 10^{-3}, 10^{-2})$

$\mu \leftarrow 1$

%% Initial value is when loop r=1

$\lambda[\ ,1] \leftarrow \mathbf{1}_n$

$\sigma_g[1] \leftarrow 1$

$\pi[1] = 0.5$

$temp[2:4] = 1/C[2:4]$

$\pi[2:4] = 0.5 \times \frac{temp[2:4]}{\sum_{k=2}^{4} temp[k]}$

%% Initial value of β_j's are estimated from a regular marginal logistic regression.

%% Initial value of μ is estimated by taking the average of the intercepts in the marginal logistic regressions.

%% We record the initial value of β_j and μ as $\beta[j,r]$ and $\mu[r]$, where $r = 1$.

%% $\beta[j,r]$ records the effect of SNP j in the rth iteration.

%% Initial value of σ_g^2 is sampled from a uniform distribution

$U = 2000$

$\sigma_g^2[1] \sim Unif(0, U)$

$\theta = 1$

$m[1:4] \leftarrow 0$

%% Finish setting the initial values

%% Start the MCMC steps:

FOR $r = 2$ to the number of MCMC iterations

%% draw values of Z and λ for each individual

FOR $i = 1$ to number of individuals n

$temp1 \leftarrow X[i,]\beta[, r-1]^T + \mu[r-1]$

%% draw $Z[i]$ from truncated logistic

$Z[i] \leftarrow logistic(temp1, 1)$

%% define $Y = 1$ as affected

IF $Y[i] = 1$

WHILE $Z[i] < 0$

$Z[i] \leftarrow logistic(temp1, 1)$

END WHILE

ELSE

WHILE $Z[i] \geq 0$

$Z[i] \leftarrow logistic(temp1, 1)$

END WHILE

END IF

%% draw new values for mixing variance

$R \leftarrow Z[i] - temp1$

$\lambda[i, r] \sim \pi(\lambda|R^2)$

%% $\lambda[i, r]$ records the λ value of individual i in the rth iteration.

%% The procedure of sampling the $\lambda[i, r]$ can be found in Holmes et al. [2006] appendix A4.

END FOR

%% draw new value of μ

$\mu[r] \sim N\left(\frac{1}{n}\sum_{i=1}^{n}(Z[i] - X[i,]\beta[, r-1]^T, \frac{1}{n^2}\sum_{i=1}^{n}\lambda[i, r]\right)$

$vara \leftarrow 0$

FOR $j = 1$ to number of SNPs p

FOR $i = 1$ to number of number of individuals n

$\tilde{Z}[i, j] \leftarrow Z[i] - \mu[r] - \sum_{l \neq j} X[i, l]\beta[l, r-1]$

END FOR

FOR $k = 1$ to number of categories (4)

IF $k = 1$

$logL[k] = log(\pi[k])$

ELSE

$logL[k] = -\frac{1}{2}\left\{log(\sum_{i=1}^{n} \frac{X[i,j]^2}{\lambda[i,r]} C[k]\sigma_g^2[r-1] + 1)\right\}$

$-\frac{1}{2}\left[- \frac{(\sum_{i=1}^{n} \frac{\tilde{Z[i,j]}X[i,j]}{\lambda[i,r]})^2}{\sum_{i=1}^{n} \frac{X[i,j]^2}{\lambda[i,r]} + \frac{1}{C[k]\sigma_g^2[r-1]}} \right] + log(\pi[k])$

END FOR

FOR $k = 1$ to number of categories (4)

$T[k] = \frac{1}{\sum_{l=1}^{4} exp(logL[l] - logL[k])}$

END FOR

$h \sim UNIF(0, 1)$

IF $h \leq T[1]$

%% draw new values for SNP effects

$\beta[j, r] = 0$

$m[1] \leftarrow m[1] + 1$

%% b[j, r] is the category SNP j belongs to

$b[j, r] = 1$

ELSE IF $h \leq T[1] + T[2]$

$\beta[j, r] \sim N\left(\frac{\sum_{i=1}^{N} \tilde{Z}[i,j]X[i,j]\lambda[i,r]^{-1}}{\sum_{i=1}^{N} \lambda[i,r]^{-1}X[i,j]^2 + (C[2]\sigma_g^2[r-1])^{-1}}, \frac{1}{\sum_{i=1}^{N} \lambda[i,r]^{-1}X[i,j]^2 + (C[2]\sigma_g^2[r-1])^{-1}}\right)$

$m[2] \leftarrow m[2] + 1$

$b[j, r] = 2$

$vara \leftarrow vara + \beta[j, r]^2/(2 \cdot C[2])$

ELSE IF $h \leq T[1] + T[2] + T[3]$

$\beta[j, r] \sim N\left(\frac{\sum_{i=1}^{N} \tilde{Z}[i,j]X[i,j]\lambda[i,r]^{-1}}{\sum_{i=1}^{N} \lambda[i,r]^{-1}X[i,j]^2 + (C[3]\sigma_g^2[r-1])^{-1}}, \frac{1}{\sum_{i=1}^{N} \lambda[i,r]^{-1}X[i,j]^2 + (C[3]\sigma_g^2[r-1])^{-1}}\right)$

$m[3] \leftarrow m[3] + 1$

$b[j, r] = 3$

$vara \leftarrow vara + \beta[j, r]^2/(2 \cdot C[3])$

ELSE

$\beta[j,r] \sim N\left(\frac{\sum_{i=1}^{N}\tilde{Z}[i,j]X[i,j]\lambda[i,r]^{-1}}{\sum_{i=1}^{N}\lambda[i,r]^{-1}X[i,j]^2+(C[4]\sigma_g^2[r-1])^{-1}}, \frac{1}{\sum_{i=1}^{N}\lambda[i,r]^{-1}X[i,j]^2+(C[4]\sigma_g^2[r-1])^{-1}}\right)$

$m[4] \leftarrow m[4]+1$

$b[j,r] = 4$

$vara \leftarrow vara + \beta[j,r]^2/(2 \cdot C[4])$

END IF

END FOR

%% Update σ_g^2 using Metropolis-Hasting sampling

$\sigma_g^{2(*)} \sim N(\sigma_g^2[r-1], \theta)$

WHILE $\sigma_g^{2(*)} \leq 0$

$\sigma_g^{2(*)} \sim N(\sigma_g^2[r-1], \theta)$

END WHILE

$RR = \frac{(\sigma_g^{2(*)})^{-\frac{1}{2}\sum_{k=2}^{4} m[k]} exp\{-\frac{1}{\sigma_g^{2(*)}}[\frac{1}{2}\sum_{k=2}^{4}(C[k]^{-1}\sum_{j:b_j=k}\beta[j,r]^2)]\}\Phi(\frac{\sigma_g^2[r-1]}{\sqrt{\theta}})}{(\sigma_g^2[r-1])^{-\frac{1}{2}\sum_{k=2}^{4} m[k]} exp\{-\frac{1}{\sigma_g^2[r-1]}[\frac{1}{2}\sum_{k=2}^{4}(C[k]^{-1}\sum_{j:b_j=k}\beta[j,r]^2)]\}\Phi(\frac{\sigma_g^{2(*)}}{\sqrt{\theta}})}$

%% Sample v from uniform distribution

$v \sim unif(0,1)$

IF $v > RR$

$\sigma_g^2[r] = \sigma_g^2[r-1]$

ELSE

$\sigma_g^2[r] = \sigma_g^{2(*)}$

%% draw a new value for $\boldsymbol{\pi}$

$\boldsymbol{\pi} \sim D(m[1]+1, m[2]+1, m[3]+1, m[4]+1)$

END

APPENDIX C

RCPP CODE OF BAYESRB R PACKAGE

The BayesRB source code can be found in the Github website:
https://github.com/sylviashanboo/BayesRB.

```
1
2
3  #include <R.h>
4  #include <Rmath.h>
5  #include <Rcpp.h>
6  #include <gsl/gsl_math.h>
7  #include <gsl/gsl_rng.h>
8  #include <gsl/gsl_sum.h>
9  #include <gsl/gsl_randist.h>
10 #include <gsl/gsl_permutation.h>
11 #include <iostream>
12 #include <math.h>
13 #include <stdio.h>
14 #include <vector>
15 #include <numeric>
16 #include <cmath>
17 #include <algorithm>
18 using namespace std;
19 using namespace Rcpp;
20
21 int rightmost(double u, double lam)
22 {
23   double z=1;
24   double x = exp(-0.5*lam);
25   int j = 0;
26   int OK;
27   while(1)
28   {
29     j ++;
30     z = z - (j+1)*(j+1)*pow(x,(j+1)*(j+1)-1);
```

```
31
32      if (z > u){
33        OK = 1;
34        break;
35      }
36      j ++;
37      z = z + (j+1)*(j+1)*pow(x,(j+1)*(j+1)-1);
38      if (z < u){
39        OK = 0;
40        break;
41      }
42    }
43    return (OK);
44  }
45
46  int leftmost(double u, double lam)
47  {
48    const double c_pi  = 3.141592653589793238463;
49    double H = 0.5*log(2) + 2.5*log(c_pi) - 2.5*log(lam)
50                          - pow(c_pi,2)/(2*lam) + 0.5*lam;
51    double lu = log(u);
52    double z=1;
53    double x = exp(-1*pow(c_pi,2)/(2*lam));
54    double k = lam/pow(c_pi,2);
55    int j =0;
56    int OK;
57    while(1){
58      j++;
59      z = z - k*pow(x,j*j-1);
60
61      if(H+log(z)>lu){
62        OK=1;
63        break;
64      }
65      j++;
66      z = z + (j+1)*(j+1)*pow(x,(j+1)*(j+1)-1);
67      if(H+log(z)<lu){
68        OK=0;
69        break;
70      }
71    }
72    return (OK);
73  }
74
75  List BayesRF(int seed, int MCMC_inte, int burn_intee, int thinn,
76  NumericMatrix X, NumericVector Y, NumericVector beta_initial,
77  double sigma2){
78
79    // generate the random seed r;
80    gsl_rng * r;
81    r = gsl_rng_alloc(gsl_rng_mt19937);
82    gsl_rng_set(r, seed);
83
84    //N is the number of individuals
```

```
85  //P is the number of SNPs
86  size_t N = X.nrow();
87    size_t P = X.ncol();
88
89  //Standardize the genotypes
90    for (int i = 0; i < P; i ++){
91      X(_,i) = (X(_,i) - mean(X(_,i)))/sd(X(_,i));
92    }
93
94  //initial values of mu and beta's:
95    NumericVector lambda(N);
96    NumericVector Z(N);
97    NumericVector beta(P);
98    for (int i =1; i<=P; i++){
99    beta[i-1] = beta_initial[i];
100 }
101   double mu=beta_initial[0];
102
103 //Set up Ck:
104   NumericVector Ck(4);
105   Ck[0] = 0;
106   Ck[1] = 0.0001;
107   Ck[2] = 0.001;
108   Ck[3] = 0.01;
109
110 //initial value of pi:
111 double* pi  = NULL;
112 pi  = new double [4];
113 double temp, sigma2k;
114 double sumup = 1/Ck[1] + 1/Ck[2] + 1/Ck[3];
115 pi[0]=0.5;
116 for (int i=1;i<=3;i++){
117   temp=1/Ck[i];
118   pi[i]=0.5*temp/sumup;
119 }
120 NumericVector m(4);
121 NumericVector xbeta_indi(N);
122 NumericMatrix Z_til(N, P);
123 NumericVector data(P);
124 for (int i=0;i<P;i++){
125   data[i] = i;
126 }
127 NumericVector bj(P);
128
129 //Set up some other variables:
130 int rest = MCMC_inte - burn_intee;
131 int mod = rest\% thinn;
132 int store_num = (rest - mod)/ thinn;
133 NumericMatrix r_beta(store_num,P);
134 NumericMatrix r_bj(store_num,P);
135 NumericMatrix r_lambda(store_num,N);
136 NumericMatrix r_Z(store_num,N);
137 NumericVector r_mu(store_num);
138 NumericMatrix r_pi(store_num,4);
```

```
139   NumericMatrix r_m(store_num,4);
140   double xbeta, all_xbeta, loct, rr,yy,uu,lambda_temp,temp_a,
141   temp_b, temp1, temp2,h,rsam, beta_pre;
142   int  snpj, num_pi, num_m;
143   NumericVector thresh(4);
144   NumericVector logL(4);
145   NumericVector beta_mu(4);
146   NumericVector beta_var(4);
147   double* temp_d  = NULL;
148   temp_d  = new double [4];
149   int num_of_store = 0;
150   cout << "# of individuals:" <<N<<"\n";
151   cout << "# of SNPs:" << P << "\n";
152   gsl_permutation * permut_p = gsl_permutation_alloc (P);
153       gsl_permutation_init (permut_p);
154
155   //Start MCMC loop:
156   for(int mcnum=1; mcnum < MCMC_inte; mcnum++){
157     cout << "mcmcnum: "<<mcnum <<"\n";
158
159     //Start MCMC step 1:
160     for(int indi=0; indi < N; indi++){
161       //calculate sum_j(X_ij*beta[j])
162       xbeta = sum(X(indi,_)*beta);
163       xbeta_indi[indi] = xbeta;
164       loct = mu + xbeta;
165       rsam = gsl_ran_logistic (r,1);
166       Z[indi] = rsam + loct;
167       if (Y[indi]==1){
168         while(Z[indi]<0){
169           rsam = gsl_ran_logistic(r,1);
170           Z[indi] = rsam + loct;
171         }
172       }
173       else{
174         while(Z[indi]>=0){
175           rsam = gsl_ran_logistic (r,1);
176           Z[indi] = rsam + loct;
177         }
178       }
179       //finish updating Z
180       ////////////////////////////////////
181       int OK=0;
182       rr = abs(rsam);
183       if(rr < 0.00001){
184         rr = 0.00001;
185       }
186       while(OK==0){
187         yy = gsl_ran_gaussian(r,1);
188         yy = yy*yy;
189         yy = 1 + (yy-sqrt(yy*(4*rr+yy)))/
190         (2*rr);
191         while (yy==-1 || yy== 0){
192           yy = gsl_ran_gaussian(r,1);
```

```
193             yy = yy*yy;
194             yy = 1 + (yy-sqrt(yy*(4*rr
195             +yy)))/(2*rr);
196           }
197           if (yy <= 0){
198             yy = 0.00001;
199           }
200           uu = gsl_rng_uniform_pos (r);
201
202           if (uu <= 1/(1+yy)){
203             lambda_temp = rr/yy;
204           }else{
205             lambda_temp = rr*yy;
206           }
207           uu = gsl_rng_uniform_pos (r);
208           if (lambda_temp > 4/3){
209             OK = rightmost(uu,lambda_temp);
210           }else{
211             OK = leftmost(uu,lambda_temp);
212           }
213         }
214         lambda[indi] = lambda_temp;
215         //finish lambda;
216       }
217       //finish loop of individual;
218       //finish step 1;
219       ////////////////////////////////////////////////
220
221       //Start MCMC step2:
222       all_xbeta = sum(xbeta_indi);
223       temp_a = (sum(Z)-all_xbeta)/N;
224       temp_b = sum(lambda)/pow(N,2);
225       mu = gsl_ran_gaussian(r,temp_b)+temp_a;
226       //finish mu;
227       //finish step 2;
228       ////////////////////////////////////////////////
229
230       //Start MCMC step 3:
231       for (num_m=0;num_m<4;num_m++){
232         m[num_m] = 0;
233       }
234       //permute the SNP order:
235       gsl_ran_shuffle (r, permut_p->data, P, sizeof(size_t));
236       for (int snpji=0; snpji<P; snpji++){
237         snpj = gsl_permutation_get(permut_p,snpji);
238         Z_til(_,snpj) = Z - mu -  xbeta_indi
239         + X(_,snpj)*beta[snpj];
240         temp1 = sum(pow(X(_,snpj),2)/lambda);
241         temp2 = sum(Z_til(_,snpj) * X(_,snpj) /lambda);
242         for (int k = 0; k < 4; k++){
243           if(k==0){
244             logL[k] = log(pi[k]);
245             beta_mu[k] = 0;
246             beta_var[k] = 0;
```

```
247           }else{
248             sigma2k = sigma2*Ck[k];
249             beta_mu[k] = temp2/(temp1
250             + 1/sigma2k);
251             beta_var[k] = 1/(temp1
252             + 1/sigma2k);
253             logL[k] = log(pi[k])
254             -0.5*log(temp1*sigma2k+1)
255             +0.5*(beta_mu[k]*temp2);
256           }
257         }
258         for (int k = 0; k < 4; k++){
259           thresh[k] = 1/sum(exp(logL - logL[k]));
260         }
261         beta_pre = beta[snpj];
262         h = gsl_rng_uniform_pos (r);
263         if (h<=thresh[0]){
264           beta[snpj] = 0;
265           m[0]+=1;
266           bj[snpj]=1;
267         }else if(h<= thresh[0] + thresh[1]){
268           beta[snpj] = gsl_ran_gaussian(r,
269           beta_var[1])+beta_mu[1];
270           m[1]+=1;
271           bj[snpj]=2;
272         }else if(h<= thresh[0] + thresh[1] +
273         thresh[2]){
274           beta[snpj] = gsl_ran_gaussian(r,
275           beta_var[2])+beta_mu[2];
276           m[2]+=1;
277           bj[snpj]=3;
278         }else{
279           beta[snpj] = gsl_ran_gaussian(r,
280           beta_var[3])+beta_mu[3];
281           m[3]+=1;
282           bj[snpj]=4;
283         }
284         xbeta_indi = xbeta_indi - X(_,snpj)
285         *beta_pre + X(_,snpj)*beta[snpj];
286       }
287       cout << "m[0,1,2,3]:"<<m[0]<<","<<m[1]
288       <<","<<m[2]<<","<<m[3]<<"\n";
289       ////////////////////////////////////////////////////
290       //finish updated beta and b_j.
291       //finish MCMC step 3.
292
293       //If set sigma_g^2 as a fixed value,
294       //then skip the MCMC step 4 shown below:
295       //Start MCMC step 4:
296       //MH sampling:
297       sigma2_temp = gsl_ran_gaussian(r,theta)+sigma2;
298       while (sigma2_temp <= 0){
299         sigma2_temp = gsl_ran_gaussian(r,theta)+sigma2;
300       }
```

```
301       cout << "sigma2_temp:" << sigma2_temp << "\n";
302       temp1 = R::pnorm(sigma2/sqrt(theta),0,1,1,0);
303       temp2 = R::pnorm(sigma2_temp/sqrt(theta),0,1,1,0);
304       logrrr = -0.5*(m[1]+m[2]+m[3])*(-log(sigma2)
305       + log(sigma2_temp)) - vara*(-1/sigma2 + 1/sigma2_temp)
306       + (log(temp1) - log(temp2));
307       cout << "logrrr:" << logrrr <<"\n";
308       u = gsl_rng_uniform_pos (r);
309       if(logrrr >= 0){
310         sigma2 = sigma2_temp;
311       }else if(exp(logrrr)>=u){
312         sigma2 = sigma2_temp;
313       }
314       if (sigma2==0){
315         cout << "break, since sigma_g^2 = 0\n";
316         break;
317       }
318       cout << "sigma2:" << sigma2 << "\n";
319       ////////////////////////////////////////////////////
320       //finished step 4: sigma_g^2.
321
322       //Start MCMC step5:
323       //updating pi
324       for(int k=0;k<4;k++){
325         temp_d[k] = m[k]+1;
326       }
327       gsl_ran_dirichlet (r,4,temp_d,pi);
328       ////////////////////////////////////////////////////
329       //finished updating pi
330       //finished MCMC step 5.
331
332       //Steps for thinning:
333       if (mcnum == burn_intee+thinn*num_of_store+thinn-1){
334         //r_beta:
335         r_beta(num_of_store,_) = beta;
336         //r_bj
337         r_bj(num_of_store,_) = bj;
338
339         //r_lambda
340         r_lambda(num_of_store,_) = lambda;
341         // r_Z:
342         r_Z(num_of_store,_) = Z;
343         //r_mu:
344         r_mu[num_of_store] = mu;
345         //r_pi
346         for(num_pi = 0; num_pi<4;num_pi++){
347           r_pi(num_of_store,num_pi) = pi[num_pi];
348         }
349         //r_m:
350         r_m(num_of_store,_) = m;
351         num_of_store++;
352       }
353       //Loop back to step 1
354     }
```

```
355
356     gsl_permutation_free (permut_p);
357     //cout << "finished mcmc loop!\n";
358     return List::create(_["r_beta"] = r_beta, _["r_bj"] = r_bj,
359     _["r_lambda"] = r_lambda, _["r_Z"] = r_Z, _["r_mu"] = r_mu,
360     _["r_pi"] = r_pi, _["r_m"] = r_m);
361 }
```

APPENDIX D

R CODE FOR THE RFGSS SIMULATION

```
1
2  library(aster)
3  library(stats)
4  library(raster)
5  library(Rlab)
6  require(ggplot2)
7  require(SPECIES)
8  require(reshape)
9  library(robustHD)
10 library("DEoptim")
11 library(doMC)
12 registerDoMC(2)
13 library(foreach)
14 library(plyr)
15
16 Args <- commandArgs()
17 set.seed(as.numeric(Args[6]))
18
19 ##some functions
20 func <- function(x, mut_r) {
21   return(rpois(1, x * mut_r*0.5))
22 }
23 g1_pass_g3 <- function(x,g2_num,num_g3_list){
24   #x is a vector (row of mut_countG1_2651) for one G1 mice
25   #selected_g2 is the number of g2 daughter each g1 mates with
26   if(sum(x)==0){
27     return((x))
28   }else{
29     n=which(x==1)
30     m=which(x>1)
31     x[m] = 1
32     n=c(n,m)
33     g3_all1=rep(0,length(x))
34     selected_g2=sample(g2_num,size=1,replace=T)
35
36     #for the genes with 1 mutations:
```

```
    for(i in 1:selected_g2){
      #calclate g2:
      g2_passed = sapply(x[n],rbinom,size=1,prob=0.5)
      #finished g2

      g3_n = sample(num_g3_list,size=1,replace=T)
      g3_matrix=matrix(rep(x,each=g3_n),ncol=g3_n,nrow=length(x),
      byrow = T)

      #one mutation on one gene
      if (sum(g2_passed + x[n] ==2)!=0){
        #if g2 have at least one mutation from g1:
        tt = which((g2_passed + x[n] ==2))
        #the number of g3 this g2 generates is:
        one_g3 = as.numeric(g2_passed + x[n]==2)
        for(j in 1:g3_n){
          g3_passed=sapply(one_g3[tt],rbinom,size=1,prob=0.25)
          g3_matrix[n[tt],j] = g3_passed
          g3_matrix[n[-tt],j] = 0
        }
      }else{
        g3_matrix=c()
      }
      g3_all1 = cbind(g3_all1,g3_matrix)
    }
  }

  unl = unlist(g3_all1)
  nr = dim(g3_all1)[1]
  nc = dim(g3_all1)[2]
  #m is the unlisted matrix of g3_all
  m = matrix(unl,nrow = nr, ncol=nc)
  dis_g3_num = apply((m),2,sum)
  if(sum(dis_g3_num)==0){
    return((rep(0,length(x))))
  }else{
    #return the diseased g3 genome
    n_temp=sample(which(dis_g3_num!=0),size=1)
    return((m[,n_temp]))
  }
}

pret_fun <- function(pretL,undet_l,obs_l,prob0,c,c1,d){
  pretn = pretL - length(obs_l)
  pret_l = sample(undet_l, pretn, replace = F, prob = prob0)
  pret_total_l = c(pret_l, obs_l)
  pret_pi = pret_total_l/sum(pret_total_l)
  pret_g = cv(pret_pi)/100
  pret_N = (C.total(c, c1, d, pret_g))$C
  pret_N_diff = pret_N - pretL
  return(list(pret_g=pret_g,pret_N=pret_N,pret_N_diff=pret_N_diff))
}

C.total <- function(c, c1, d, g) {
```

```
  # c is the number of observed risk-associated genes
  # c1 is the number of genes mutated once
  # d is the total number of risk-associated mutations
  # g is the variation in effect size of individual de novo mutations
  u <- 1 - c1/d
  # probability that newly added mutation hits a previously mutated
  # gene
  C <- c/u + g^2 * d * (1 - u)/u
  zero_class <- (C - c)/C
  return(list(C = C, zero_class = zero_class))
}
load("/ihome/dweeks/yis29/mice_project/family_revised0425/tal_l.Rdata")
n_rep=2
#n_rep=2
mut_rate = 1/(1.82 * 1e+06)
p_detect = 90/113
#n_sim=100
n_sim=100
#nmice is the number of G3 mice that one G0 mice generates.

fam = read.csv("/ihome/dweeks/yis29/mice_project/family_revised0425/
YingLineSummary.csv",header = T)
fath_num=length(unique(fam[,2]))
moth_num=ddply(fam,.(Mother.ID),nrow)[,2]
g3_num = moth_num[which(moth_num>1)]
g2_num_temp = ddply(fam,.(Father.ID,Mother.ID),nrow)[,c(1,2)]
g2_num = ddply(g2_num_temp,.(Father.ID),nrow)[,2]

LL <- seq(400, 800, by = 200)
result_arr = array(NA, dim = c(14, n_rep, length(LL)),
dimnames = c("method", "nrep", "L"))
new_method_simu = array(NA, dim = c(3, n_rep, length(LL)),
dimnames = c("quantile", "nrep", "L"))


list_result<-foreach(fech_num=1:4) \%dopar\%{
  nL = 0
  for (L in LL) {
    nL <- nL + 1
    rep = 0
    repeat {
      time0 <- proc.time()
      total_l <- sample(tal_l, L, replace = F)
      # Select 113 mice with one or more mutations for exome sequencing
      mut_countG1_400 <- matrix(nrow = L, ncol = 800)
      for (nmice in 1:800) {
        mut_countG1_400[,nmice]<- sapply(total_l,
        func,mut_r = mut_rate)
      }
      mut_countALL = apply(mut_countG1_400, 2, g1_pass_g3, g2_num,
      g3_num)

      mut_countG1_100 <- (matrix(nrow = L, ncol = 100))
      while(sum(colSums(mut_countALL) > 0) < 113){
```

```
145         #mutations in G1
146         for (nmice in 1:100) {
147           mut_countG1_100[,nmice]<- sapply(total_l,func,mut_r=mut_rate)
148         }
149         mut_countALL100=(apply(mut_countG1_100, 2, g1_pass_g3, g2_num,
150         g3_num))
151         mut_countALL = cbind(mut_countALL, mut_countALL100)
152       }
153       rep = rep + 1
154
155       n113 <- sample(which(colSums(mut_countALL) > 0), size = 113,
156       replace = FALSE)
157       #length(which(rowSums(mut_countALL) > 0))
158       # detection step:
159       m113 <- data.frame(matrix(NA, nrow = 113, ncol = L))
160       j = 0
161       for (i in n113) {
162         j = j + 1
163         temp = mut_countALL[,i]
164         for (kk in which(temp != 0)) {
165           temp[kk] = sum(sapply(temp[kk], rbinom, size = 1,
166           p = p_detect))
167         }
168         m113[j, ] = temp
169       }
170       # after the detection step. The observed matrix.
171       names(m113) <- total_l
172       n_obs_col = which(colSums(m113) > 0)
173
174       obs_maxtr = m113[, n_obs_col]
175       # obs_maxtr is a matrix that we can observed
176       # the rows of the matrix is observed mice
177       # the columns of the matrix is the observed CHD genes
178       # there are 90 columns of the 'obs_maxtr' in Dr. Lo's experiment
179
180       # obs_l is the length of the observed gene length
181       obs_l = as.numeric(names(n_obs_col))
182
183
184       ###################################
185
186       ############################
187
188       ############################
189
190       # undet_l is the gene length of all the genes other than the
191       # observed genes in the mice genome
192       undet_l <- tal_l
193       for (aa in 1:length(obs_l)) {
194         undet_l <- undet_l[-((which(undet_l == obs_l[aa]))[1])]
195       }
196
197       ##################
198       # unseen species
```

```
199         c = length(obs_maxtr[1, ])
200         c1 = sum(apply(obs_maxtr, 2, sum) == 1)
201         d = sum(apply(obs_maxtr, 2, sum))
202         u = 1 - c1/d
203         cat = seq(max(200,length(n_obs_col)), max(3000,
204         length(n_obs_col)), by = 20)
205         est_N_estim3 = rep(NA, n_sim)
206         nnrep = 0
207         index=1:length(cat)
208
209         #Start the proposed method:
210         mean_n2 = mean(g2_num)
211         mean_n3 = mean(g3_num)
212         #pp_inheri = 1-P(G1 is not muated) - (P(G2 is not mutated)
213         #+ P(G2 is mutated)*P(G2 is not homozygous)^(#ofG3))^(#ofG2)
214         pp_inheri=1-0.5-(0.5*(0.75)^mean_n3 + 0.5)^mean_n2
215         #it is proofed that this step is equivalent to poisson+detection
216         prob0=(dpois(0, undet_l * mut_rate  * p_detect * pp_inheri))^113
217         repeat{
218           pret_mat= sapply(cat,pret_fun,undet_l,obs_l,prob0,c,c1,d)
219           colnames(pret_mat) =  cat
220           pret_N_diff = unlist(pret_mat[3,])
221           nnrep=nnrep+1
222           est_N_estim3[nnrep] = as.numeric(names(pret_N_diff[which.min(
223           abs(pret_N_diff))]))
224           print(paste("finished small loop:",nnrep))
225           if(nnrep >= n_sim){
226             break
227           }
228
229         } # Loop on nnrep in 1:100
230         result_arr[1, rep, nL] = mean(est_N_estim3, na.rm = T)
231         est_q = quantile(est_N_estim3, c(0.05, 0.5, 0.95), na.rm = T)
232         # we use the 5, 95 quantile to construct a interval
233         result_arr[2, rep, nL] = est_q[1]
234         result_arr[3, rep, nL] = est_q[2]
235         result_arr[4, rep, nL] = est_q[3]
236
237         ########
238         ## The other 7 non-parametric methods:
239         data = data.frame(tabulate(apply(obs_maxtr, 2, sum)))
240         data$j <- as.integer(rownames(data))
241         data <- data[, c(2, 1)]
242         names(data) <- c("j", "n_j")
243         est_N_ChaoLee1992 <- ChaoLee1992(data)$Nhat[2]
244         result_arr[5, rep, nL] <- est_N_ChaoLee1992
245         est_N_jackknife <- jackknife(data)$Nhat
246         result_arr[6, rep, nL] <- est_N_jackknife
247         est_N_Chao1984 <- chao1984(data)$Nhat
248         result_arr[7, rep, nL] <- est_N_Chao1984
249         est_N_ChaoBunge <- ChaoBunge(data, t = 10)$Nhat
250         result_arr[8, rep, nL] <- est_N_ChaoBunge
251         est_N_pnpmle <- pnpmle(data, t = 15, C = 0, b = 200)$Nhat
252         result_arr[9, rep, nL] <- est_N_pnpmle
```

```
253         est_N_unpmle <- unpmle(data, t = 15, C = 0, b = 200)$Nhat
254         result_arr[10, rep, nL] <- est_N_unpmle
255         result_arr[11, rep, nL] <- (C.total(c, c1, d, 1))$C
256
257         ##parametric methods:
258         temp = rep(0,100)
259         names(temp) = 1:100
260         temptab=table(apply(obs_maxtr,2,sum))
261         nametemptab = names(temptab)
262         temp[nametemptab] = temptab
263         for(i in 1:100){
264           if(sum(temp[(i+1):100])==0){
265             paper2 = temp[1:i]
266             break
267           }
268         }
269         paper1 = as.numeric(names(paper2))
270         alloci <- sum(paper2)
271         paper <- rbind(paper1,paper2)
272         papernum <- c()
273         for (i in 1:length(paper2)){
274           papernum <- c(papernum,rep(paper1[i],times=paper2[i]))
275         }
276         # Truncated Poisson distribution:
277         poislnL <- function(lam){-1*log(1 - (1/exp(lam)))
278         *length(papernum)+ sum(log(dpois(papernum,lam))) }
279         x <- seq(0.1, 10, len=500)
280         y <- sapply(x,poislnL)
281         fitpoi <- optimize(poislnL, c(0, 1000),maximum = T)
282         #fitpoi
283         alloci <- sum(paper2)
284         poisy1 <- dpois(paper1,fitpoi$maximum)
285         poiallnum <- alloci/sum(poisy1)
286         result_arr[12, rep, nL] <- poiallnum
287
288         # Truncated negative binomial distribution:
289         #f <- function(k){
290           # the function is written according to the paper
291         #  a <- k[1]
292         #  b <- k[2]
293         #  z <- ((b+1)^(-1*a))*factorial(papernum+a-1)/
294         (factorial(papernum)*factorial(a-1))*((b/(b+1))^papernum)
295         #  sum(log(z))-log(1-(b+1)^(-1*a))*length(papernum)
296         #}
297         #fitnega <- optim(c(1,1), f, control=list(fnscale=-1),
298         lower=c(0.001,0.0001), upper=c(10,10000), method="L-BFGS-B")
299         #a <- fitnega$par[1]
300         #b <- fitnega$par[2]
301         #nby1 <- ((b+1)^(-1*a))*factorial(paper1+a-1)/(factorial(paper1)
302         *factorial(a-1))*((b/(b+1))^paper1)
303         #nballnum <- alloci/sum(nby1)
304         ##nballnum is too unrealistic. So, we wont use this method.
305
306         ##2C method:
```

```
307         poislnL2 <- function(lam){
308           lam1 <- lam[1]
309           lam2 <- lam[2]
310 
311           sum(log(
312             dpois(papernum,lam1)/2/(1-(1/exp(lam1)))
313             +dpois(papernum,lam2)/2/(1-(1/exp(lam2)))
314           ))
315 
316         }
317         fitpoi2 <- optim(c(1,3), poislnL2, control=list(fnscale=-1),
318         lower=c(1e-15,0.01),upper=c(10,10000), method="L-BFGS-B")
319         lam1=fitpoi2$par[1]
320         lam2=fitpoi2$par[2]
321         c2y1 <- dpois(paper1,lam1)/2+dpois(paper1,lam2)/2
322         c2allnum <- alloci/sum(c2y1)
323         result_arr[13, rep, nL] <- c2allnum
324 
325         #3C model:
326         poislnL3 <- function(lam){
327           lam1 <- lam[1]
328           lam2 <- lam[2]
329           lam3 <- lam[3]
330           sum(log(
331             dpois(papernum,lam1)/3/(1-(1/exp(lam1)))
332             +dpois(papernum,lam2)/3/(1-(1/exp(lam2)))
333             +dpois(papernum,lam3)/3/(1-(1/exp(lam3)))
334           ))
335 
336         }
337         fitpoi3 <- optim(c(1,3,5), poislnL3, control=list(fnscale=-1),
338         lower=c(0.01,0.01,0.01), upper=c(10,10000,10000),
339         method="L-BFGS-B")
340         lam1=fitpoi3$par[1]
341         lam2=fitpoi3$par[2]
342         lam3=fitpoi3$par[3]
343         c3y1 <- dpois(paper1,lam1)/3+dpois(paper1,lam2)/3
344         +dpois(paper1,lam3)/3
345         c3allnum <- alloci/sum(c3y1)
346         result_arr[14, rep, nL] <- c3allnum
347 
348         if (rep >= n_rep) {
349           break
350         }
351         print(paste("finished big loop:",rep,sep=" "))
352         proc.time() - time0
353       } # repeat up to n_rep
354     } # Loop on L in LL
355 
356     result_arr
357 }#foreach
358 #list_result
359 
360 sf_name = paste("/ihome/dweeks/yis29/mice_project/family_revised0425/
```

```
361 list_result",as.numeric(Args[6]),".Rdata",sep="")
362 save(list_result,file=sf_name)
```

BIBLIOGRAPHY

Albert, J. H. and Chib, S. (1993). Bayesian analysis of binary and polychotomous response data. Journal of the American statistical Association *88*, 669–679.

Assar, A. N. and Zarins, C. K. (2009). Ruptured abdominal aortic aneurysm: a surgical emergency with many clinical presentations. Postgrad Med J *85*, 268–73.

Biros, E., Norman, P. E., Jones, G. T., van Rij, A. M., Yu, G., Moxon, J. V., Blankensteijn, J. D., van Sterkenburg, S. M., Morris, D., Baas, A. F. and Golledge, J. (2011). Meta-analysis of the association between single nucleotide polymorphisms in TGF-beta receptor genes and abdominal aortic aneurysm. Atherosclerosis *219*, 218–23.

Bloss, C. S., Darst, B. F., Topol, E. J. and Schork, N. J. (2011). Direct-to-consumer personalized genomic testing. Human Molecular Genetics *1*, ddr349.

Borthwick, K. M., Smelser, D. T., Bock, J. A., Elmore, J. R., Ryer, E. J., Ye, Z., Pacheco, J. A., Carrell, D. S., Michalkiewicz, M., Thompson, W. K. et al. (2015). Ephenotyping for abdominal aortic aneurysm in the electronic medical records and genomics (emerge) network: algorithm development and Konstanz Information Miner Workflow. International journal of biomedical data mining *4*.

Bowden, J. and Dudbridge, F. (2009). Unbiased estimation of odds ratios: combining genomewide association scans with replication studies. Genetic epidemiology *33*, 406–418.

Bown, M. J., Jones, G. T., Harrison, S. C., Wright, B. J., Bumpstead, S., Baas, A. F., Gretarsdottir, S., Badger, S. A., Bradley, D. T., Burnand, K., Child, A. H., Clough, R. E., Cockerill, G., Hafez, H., Scott, D. J., Futers, S., Johnson, A., Sohrabi, S., Smith, A., Thompson, M. M., van Bockxmeer, F. M., Waltham, M., Matthiasson, S. E., Thorleifsson, G., Thorsteinsdottir, U., Blankensteijn, J. D., Teijink, J. A., Wijmenga, C., de Graaf, J., Kiemeney, L. A., Assimes, T. L., McPherson, R., Consortium, C. A., Global, B. C., Consortium, D., Consortium, V., Folkersen, L., Franco-Cereceda, A., Palmen, J., Smith, A. J., Sylvius, N., Wild, J. B., Refstrup, M., Edkins, S., Gwilliam, R., Hunt, S. E., Potter, S., Lindholt, J. S., Frikke-Schmidt, R., Tybjaerg-Hansen, A., Hughes, A. E., Golledge, J., Norman, P. E., van Rij, A., Powell, J. T., Eriksson, P., Stefansson, K., Thompson, J. R., Humphries, S. E., Sayers, R. D., Deloukas, P. and Samani, N. J. (2011). Abdominal aortic

aneurysm is associated with a variant in low-density lipoprotein receptor-related protein 1. Am J Hum Genet *89*, 619–27.

Browning, B. L. and Browning, S. R. (2016). Genotype imputation with millions of reference samples. The American Journal of Human Genetics *98*, 116–126.

Burnham, K. P. and Overton, W. S. (1978). Estimation of the Size of a Closed Population When Capture Probabilities Vary Among Animals. Biometrika *65*, 625–633.

Burnham, K. P. and Overton, W. S. (1979). Robust Estimation of Population Size When Capture Probabilities Vary Among Animals. Ecology *60*, 927–936.

Burton, P. R., Clayton, D. G., Cardon, L. R., Craddock, N., Deloukas, P., Duncanson, A., Kwiatkowski, D. P., McCarthy, M. I., Ouwehand, W. H., Samani, N. J. et al. (2007). Genome-wide association study of 14,000 cases of seven common diseases and 3,000 shared controls. Nature *447*, 661–678.

Chao, A. (1984). Nonparametric Estimation of the Number of Classes in a Population. Scandinavian Journal of Statistics *11*, 265–270.

Chao, A. and Bunge, J. (2002). Estimating the number of species in a stochastic abundance model. Biometrics *58*, 531–539.

Chao, A. and Lee, S.-M. (1992). Estimating the number of classes via sample coverage. American Statistical Association *87*, 210–17.

Chen, M.-H. and Dey, D. K. (1998). Bayesian Modeling of Correlated Binary Responses via Scale Mixture of Multivariate Normal Link Functions. The Indian Journal of Statistics *60*.

Crouch, D. J., Goddard, G. H. and Lewis, C. M. (2013). REGENT: a risk assessment and classification algorithm for genetic and environmental factors. Eur J Hum Genet *21*, 109–11.

De Jager, P. L., Chibnik, L. B., Cui, J., Reischl, J., Lehr, S., Simon, K. C., Aubin, C., Bauer, D., Heubach, J. F., Sandbrink, R., Tyblova, M., Lelkova, P., Havrdova, E., Pohl, C., Horakova, D., Ascherio, A., Hafler, D. A. and Karlson, E. W. (2009). Integration of genetic risk factors into a clinical algorithm for multiple sclerosis susceptibility: a weighted genetic risk score. The Lancet. Neurology *8*, 1111–9.

Elmore, J. R., Obmann, M. A., Kuivaniemi, H., Tromp, G., Gerhard, G. S., Franklin, D. P., Boddy, A. M. and Carey, D. J. (2009). Identification of a genetic variant associated with abdominal aortic aneurysms on chromosome 3p12.3 by genome wide association. J Vasc Surg *49*, 1525–31.

Erbe, M., Hayes, B., Matukumalli, L., Goswami, S., Bowman, P., Reich, C., Mason, B. and Goddard, M. (2012). Improving accuracy of genomic predictions within and between dairy

cattle breeds with imputed high-density single nucleotide polymorphism panels. Journal of dairy science *95*, 4114–4129.

Evans, D. M., Visscher, P. M. and Wray, N. R. (2009). Harnessing the information contained within genome-wide association studies to improve individual prediction of complex disease risk. Human molecular genetics *18*, 3525–3531.

Feng, Z. Z., Yang, X., Subedi, S. and McNicholas, P. D. (2012). The LASSO and sparse least square regression methods for SNP selection in predicting quantitative traits. IEEE/ACM transactions on computational biology and bioinformatics / IEEE, ACM *9*, 629–36.

Fritsche, L. G., Chen, W., Schu, M., Yaspan, B. L., Yu, Y., Thorleifsson, G., Zack, D. J., Arakawa, S., Cipriani, V., Ripke, S., Igo, R. P., J., Buitendijk, G. H., Sim, X., Weeks, D. E., Guymer, R. H., Merriam, J. E., Francis, P. J., Hannum, G., Agarwal, A., Armbrecht, A. M., Audo, I., Aung, T., Barile, G. R., Benchaboune, M., Bird, A. C., Bishop, P. N., Branham, K. E., Brooks, M., Brucker, A. J., Cade, W. H., Cain, M. S., Campochiaro, P. A., Chan, C. C., Cheng, C. Y., Chew, E. Y., Chin, K. A., Chowers, I., Clayton, D. G., Cojocaru, R., Conley, Y. P., Cornes, B. K., Daly, M. J., Dhillon, B., Edwards, A. O., Evangelou, E., Fagerness, J., Ferreyra, H. A., Friedman, J. S., Geirsdottir, A., George, R. J., Gieger, C., Gupta, N., Hagstrom, S. A., Harding, S. P., Haritoglou, C., Heckenlively, J. R., Holz, F. G., Hughes, G., Ioannidis, J. P., Ishibashi, T., Joseph, P., Jun, G., Kamatani, Y., Katsanis, N., C, N. K., Khan, J. C., Kim, I. K., Kiyohara, Y., Klein, B. E., Klein, R., Kovach, J. L., Kozak, I., Lee, C. J., Lee, K. E., Lichtner, P., Lotery, A. J., Meitinger, T., Mitchell, P., Mohand-Said, S., Moore, A. T., Morgan, D. J., Morrison, M. A., Myers, C. E., Naj, A. C., Nakamura, Y., Okada, Y., Orlin, A., Ortube, M. C., Othman, M. I., Pappas, C., Park, K. H., Pauer, G. J., Peachey, N. S., Poch, O., Priya, R. R., Reynolds, R., Richardson, A. J., Ripp, R., Rudolph, G., Ryu, E. et al. (2013). Seven new loci associated with age-related macular degeneration. Nat Genet *45*, 433–9, 439e1–2.

Galora, S., Saracini, C., Palombella, A. M., Pratesi, G., Pulli, R., Pratesi, C., Abbate, R. and Giusti, B. (2013). Low-density lipoprotein receptor-related protein 5 gene polymorphisms and genetic susceptibility to abdominal aortic aneurysm. J Vasc Surg *58*, 1062–8 e1.

Gamerman, D. (1997). Sampling from the posterior distribution in generalized linear mixed models. Statistics and Computing *7*, 57–68.

Gianola, D. (2013). Priors in whole-genome regression: the Bayesian alphabet returns. Genetics *194*, 573–596.

Giusti, B., Saracini, C., Bolli, P., Magi, A., Sestini, I., Sticchi, E., Pratesi, G., Pulli, R., Pratesi, C. and Abbate, R. (2008). Genetic analysis of 56 polymorphisms in 17 genes involved in methionine metabolism in patients with abdominal aortic aneurysm. J Med Genet *45*, 721–30.

Goddard, G. H. and Lewis, C. M. (2010). Risk categorization for complex disorders according to genotype relative risk and precision in parameter estimates. Genet Epidemiol *34*, 624–32.

Gogarten, S. M., Bhangale, T., Conomos, M. P., Laurie, C. A., McHugh, C. P., Painter, I., Zheng, X., Crosslin, D. R., Levine, D., Lumley, T. et al. (2012). GWASTools: an R/Bioconductor package for quality control and analysis of Genome-Wide Association Studies. Bioinformatics *28*, 3329–3331.

Good, I. (1953). The population frequencies of species and the estimation of population parameters. Biometrika *40*, 237–264.

Han, P. K. (2013). Conceptual, methodological, and ethical problems in communicating uncertainty in clinical evidence. Medical Care Research and Review *70*, 14S–36S.

Han, P. K., Klein, W. M., Lehman, T., Killam, B., Massett, H. and Freedman, A. N. (2011). Communication of uncertainty regarding individualized cancer risk estimates effects and influential factors. Medical Decision Making *31*, 354–366.

Harrison, S. C., Smith, A. J., Jones, G. T., Swerdlow, D. I., Rampuri, R., Bown, M. J., Aneurysm, C., Folkersen, L., Baas, A. F., de Borst, G. J., Blankensteijn, J. D., Price, J. F., van der Graaf, Y., McLachlan, S., Agu, O., Hofman, A., Uitterlinden, A. G., Franco-Cereceda, A., Ruigrok, Y. M., van't Hof, F. N., Powell, J. T., van Rij, A. M., Casas, J. P., Eriksson, P., Holmes, M. V., Asselbergs, F. W., Hingorani, A. D. and Humphries, S. E. (2013). Interleukin-6 receptor pathways in abdominal aortic aneurysm. Eur Heart J *34*, 3707–16.

Hart, S. D. and Cooke, D. J. (2013). Another look at the (Im-) precision of individual risk estimates made using actuarial risk assessment instruments. Behavioral sciences & the law *31*, 81–102.

Helgadottir, A., Gretarsdottir, S., Thorleifsson, G., Holm, H., Patel, R. S., Gudnason, T., Jones, G. T., van Rij, A. M., Eapen, D. J., Baas, A. F., Tregouet, D. A., Morange, P. E., Emmerich, J., Lindblad, B., Gottsater, A., Kiemeny, L. A., Lindholt, J. S., Sakalihasan, N., Ferrell, R. E., Carey, D. J., Elmore, J. R., Tsao, P. S., Grarup, N., Jorgensen, T., Witte, D. R., Hansen, T., Pedersen, O., Pola, R., Gaetani, E., Magnadottir, H. B., Wijmenga, C., Tromp, G., Ronkainen, A., Ruigrok, Y. M., Blankensteijn, J. D., Mueller, T., Wells, P. S., Corral, J., Soria, J. M., Souto, J. C., Peden, J. F., Jalilzadeh, S., Mayosi, B. M., Keavney, B., Strawbridge, R. J., Sabater-Lleal, M., Gertow, K., Baldassarre, D., Nyyssonen, K., Rauramaa, R., Smit, A. J., Mannarino, E., Giral, P., Tremoli, E., de Faire, U., Humphries, S. E., Hamsten, A., Haraldsdottir, V., Olafsson, I., Magnusson, M. K., Samani, N. J., Levey, A. I., Markus, H. S., Kostulas, K., Dichgans, M., Berger, K., Kuhlenbaumer, G., Ringelstein, E. B., Stoll, M., Seedorf, U., Rothwell, P. M., Powell, J. T., Kuivaniemi, H., Onundarson, P. T., Valdimarsson, E., Matthiasson, S. E., Gudbjartsson, D. F., Thorgeirsson, G., Quyyumi, A. A., Watkins, H., Farrall, M., Thorsteinsdottir, U. and Stefansson, K. (2012). Apolipoprotein(a) genetic sequence variants associated with systemic atherosclerosis and coronary atherosclerotic burden but not with venous thromboembolism. J Am Coll Cardiol *60*, 722–9.

Holmes, C. C. and Held, L. (2006). Bayesian auxiliary variable models for binary and multinomial regression. Bayesian Analysis *1*, 145–168

Howie, B. N., Donnelly, P. and Marchini, J. (2009). A flexible and accurate genotype imputation method for the next generation of genome-wide association studies. PLoS Genet *5*, e1000529.

Ionita-Laza, I., Lange, C. and Laird, N. M. (2009). Estimating the number of unseen variants in the human genome. Proceedings of the National Academy of Sciences *106*, 5008–5013.

Jones, G. T., Bown, M. J., Gretarsdottir, S., Romaine, S. P., Helgadottir, A., Yu, G., Tromp, G., Norman, P. E., Jin, C., Baas, A. F., Blankensteijn, J. D., Kullo, I. J., Phillips, L. V., Williams, M. J., Topless, R., Merriman, T. R., Vasudevan, T. M., Lewis, D. R., Blair, R. D., Hill, A. A., Sayers, R. D., Powell, J. T., Deloukas, P., Thorleifsson, G., Matthiasson, S. E., Thorsteinsdottir, U., Golledge, J., Ariens, R. A., Johnson, A., Sohrabi, S., Scott, D. J., Carey, D. J., Erdman, R., Elmore, J. R., Kuivaniemi, H., Samani, N. J., Stefansson, K. and van Rij, A. M. (2013). A sequence variant associated with sortilin-1 (SORT1) on 1p13.3 is independently associated with abdominal aortic aneurysm. Hum Mol Genet *22*, 2941–7.

Kalf, R. R., Mihaescu, R., Kundu, S., de Knijff, P., Green, R. C. and Janssens, A. C. (2014). Variations in predicted risks in personal genome testing for common complex diseases. Genet Med *16*, 85–91.

Kampstra, P. (2008). Beanplot: A Boxplot Alternative for Visual Comparison of Distributions. Journal of Statistical Software *28*.

Klein, R., Chou, C. F., Klein, B. E., Zhang, X., Meuer, S. M. and Saaddine, J. B. (2011). Prevalence of age-related macular degeneration in the US population. Arch Ophthalmol *129*, 75–80.

Kumasaka, N., Fujisawa, H., Hosono, N., Okada, Y., Takahashi, A., Nakamura, Y., Kubo, M. and Kamatani, N. (2011). PlatinumCNV: a Bayesian Gaussian mixture model for genotyping copy number polymorphisms using SNP array signal intensity data. Genetic epidemiology *35*, 831–844.

Lautenbach, D. M., Christensen, K. D., Sparks, J. A. and Green, R. C. (2013). Communicating genetic risk information for common disorders in the era of genomic medicine. Annual review of genomics and human genetics *14*, 491–513.

Li, C. and Li, M. (2008). GWAsimulator: a rapid whole-genome simulation program. Bioinformatics *24*, 140–142.

Li, Y., Klena, N. T., Gabriel, G. C., Liu, X., Kim, A. J., Lemke, K., Chen, Y., Chatterjee, B., Devine, W., Damerla, R. R., Chang, C., Yagi, H., San Agustin, J. T., Thahir, M., Anderton, S., Lawhead, C., Vescovi, A., Pratt, H., Morgan, J., Haynes, L., Smith, C. L., Eppig, J. T., Reinholdt, L., Francis, R., Leatherbury, L., Ganapathiraju, M. K., Tobita,

K., Pazour, G. J. and Lo, C. W. (2015). Global genetic analysis in mice unveils central role for cilia in congenital heart disease. Nature *521*, 520–4.

Liu, J. Z., van Sommeren, S., Huang, H., Ng, S. C., Alberts, R., Takahashi, A., Ripke, S., Lee, J. C., Jostins, L., Shah, T. et al. (2015). Association analyses identify 38 susceptibility loci for inflammatory bowel disease and highlight shared genetic risk across populations. Nature genetics *47*, 979–986.

McGeechan, K., Macaskill, P., Irwig, L. and Bossuyt, P. M. (2014). An assessment of the relationship between clinical utility and predictive ability measures and the impact of mean risk in the population. BMC Med Res Methodol *14*, 86.

Morrison, A. C., Bare, L. A., Chambless, L. E., Ellis, S. G., Malloy, M., Kane, J. P., Pankow, J. S., Devlin, J. J., Willerson, J. T. and Boerwinkle, E. (2007). Prediction of coronary heart disease risk using a genetic risk score: the Atherosclerosis Risk in Communities Study. American journal of epidemiology *166*, 28–35.

Moser, G., Lee, S. H., Hayes, B. J., Goddard, M. E., Wray, N. R. and Visscher, P. M. (2015). Simultaneous discovery, estimation and prediction analysis of complex traits using a bayesian mixture model. PLoS genetics *11*, e1004969.

Mukhopadhyay, N., Almasy, L., Schroeder, M., Mulvihill, W. P. and Weeks, D. E. (2005). Mega2: data-handling for facilitating genetic linkage and association analyses. Bioinformatics *21*, 2556–2557.

Norris, J. L. and Pollock, K. H. (1996). Nonparametric MLE under two closed capture-recapture models with heterogeneity. Biometrics *1*, 639–649.

Norris, J. L. and Pollock, K. H. (1998). Non-parametric MLE for Poisson species abundance models allowing for heterogeneity between species. Environmental and Ecological Statistics *5*, 391–402.

Pepe, M. S., Gu, J. W. and Morris, D. E. (2010a). The potential of genes and other markers to inform about risk. Cancer Epidemiol Biomarkers Prev *19*, 655–65.

Pepe, M. S., Gu, J. W. and Morris, D. E. (2010b). The potential of genes and other markers to inform about risk. Cancer Epidemiology Biomarkers & Prevention *19*, 655–665.

Pollock, D. D. and Larkin, J. C. (2004). Estimating the degree of saturation in mutant screens. Genetics *168*, 489–502.

Polson, N. G., Scott, J. G. and Windle, J. (2013). Bayesian inference for logistic models using Pólya–Gamma latent variables. Journal of the American Statistical Association *108*, 1339–1349

Prentice, R. L. and Pyke, R. (1979). Logistic disease incidence models and case-control studies. Biometrika *66*, 403–411.

Pruim, R. J., Welch, R. P., Sanna, S., Teslovich, T. M., Chines, P. S., Gliedt, T. P., Boehnke, M., Abecasis, G. R. and Willer, C. J. (2010). LocusZoom: regional visualization of genome-wide association scan results. Bioinformatics *26*, 2336–2337.

Purcell, S., Neale, B., Todd-Brown, K., Thomas, L., Ferreira, M. A., Bender, D., Maller, J., Sklar, P., De Bakker, P. I., Daly, M. J. et al. (2007). PLINK: a tool set for whole-genome association and population-based linkage analyses. The American Journal of Human Genetics *81*, 559–575.

Reese, S. E., Archer, K. J., Therneau, T. M., Atkinson, E. J., Vachon, C. M., De Andrade, M., Kocher, J.-P. A. and Eckel-Passow, J. E. (2013). A new statistic for identifying batch effects in high-throughput genomic data that uses guided principal components analysis. Bioinformatics *29*, btt480.

Rooke, T. W., Hirsch, A. T., Misra, S., Sidawy, A. N., Beckman, J. A., Findeiss, L. K., Golzarian, J., Gornik, H. L., Halperin, J. L., Jaff, M. R. et al. (2011). 2011 ACCF/AHA focused update of the guideline for the management of patients with peripheral artery disease (updating the 2005 guideline): a report of the American College of Cardiology Foundation/American Heart Association Task Force on Practice Guidelines. Journal of the American College of Cardiology *58*, 2020–2045.

Sanders, S. J., Murtha, M. T., Gupta, A. R., Murdoch, J. D., Raubeson, M. J., Willsey, A. J., Ercan-Sencicek, A. G., DiLullo, N. M., Parikshak, N. N., Stein, J. L., Walker, M. F., Ober, G. T., Teran, N. A., Song, Y., El-Fishawy, P., Murtha, R. C., Choi, M., Overton, J. D., Bjornson, R. D., Carriero, N. J., Meyer, K. A., Bilguvar, K., Mane, S. M., Sestan, N., Lifton, R. P., Günel, M., Roeder, K., Geschwind, D. H., Devlin, B. and State, M. W. (2012). De novo mutations revealed by whole-exome sequencing are strongly associated with autism. Nature *485*, 237–41.

Saracini, C., Bolli, P., Sticchi, E., Pratesi, G., Pulli, R., Sofi, F., Pratesi, C., Gensini, G. F., Abbate, R. and Giusti, B. (2012). Polymorphisms of genes involved in extracellular matrix remodeling and abdominal aortic aneurysm. Journal of vascular surgery *55*, 171–179.

Scott, I. C., Seegobin, S. D., Steer, S., Tan, R., Forabosco, P., Hinks, A., Eyre, S., Morgan, A. W., Wilson, A. G., Hocking, L. J., Wordsworth, P., Barton, A., Worthington, J., Cope, A. P. and Lewis, C. M. (2013). Predicting the risk of rheumatoid arthritis and its age of onset through modelling genetic risk variants with smoking. PLoS Genet *9*, e1003808.

Shan, Y., Smelser, D., Tromp, G., Kuivaniemi, H. and Weeks, D. (2016). Genetic risk models: model size and confidence intervals of the risk estimates. Submitted to Genetic Epidemiology.

Smelser, D. T., Tromp, G., Elmore, J. R., Kuivaniemi, H., Franklin, D. P., Kirchner, H. L. and Carey, D. J. (2014). Population risk factor estimates for abdominal aortic aneurysm from electronic medical records: a case control study. BMC Cardiovasc Disord *14*, 174.

Staples, J., Qiao, D., Cho, M. H., Silverman, E. K., Nickerson, D. A., Below, J. E., of Washington Center for Mendelian Genomics, U. et al. (2014). PRIMUS: rapid reconstruction of pedigrees from genome-wide estimates of identity by descent. The American Journal of Human Genetics *95*, 553–564.

van Dieren, S., Beulens, J. W. J., Kengne, A. P., Peelen, L. M., Rutten, G. E. H. M., Woodward, M., van der Schouw, Y. T. and Moons, K. G. M. (2012). Prediction models for the risk of cardiovascular disease in patients with type 2 diabetes: a systematic review. Heart (British Cardiac Society) *98*, 360–9.

Verma, S. S., de Andrade, M., Tromp, G., Kuivaniemi, H., Pugh, E., Namjou-Khales, B., Mukherjee, S., Jarvik, G. P., Kottyan, L. C., Burt, A., Bradford, Y., Armstrong, G. D., Derr, K., Crawford, D. C., Haines, J. L., Li, R., Crosslin, D. and Ritchie, M. D. (2014). Imputation and quality control steps for combining multiple genome-wide datasets. Front Genet *5*, 370.

Wang, J.-P. and Lindsay, B. G. (2008). An exponential partial prior for improving nonparametric maximum likelihood estimation in mixture models. Statistical Methodology *5*, 30–45.

Wang, J.-P. et al. (2011). SPECIES: an R package for species richness estimation. Journal of Statistical Software *40*, 1–15.

Wang, J.-P. Z. and Lindsay, B. G. (2005). A penalized nonparametric maximum likelihood approach to species richness estimation. Journal of the American Statistical Association *100*, 942–959.

Weeks, D. E., Conley, Y. P., Mah, T. S., Paul, T. O., Morse, L., Ngo-Chang, J., Dailey, J., Ferrell, R. E. and Gorin, M. B. (2000). A full genome scan for age-related maculopathy. Human molecular genetics *9*, 1329–1349.

Weeks, D. E., Conley, Y. P., Tsai, H.-J., Mah, T. S., Schmidt, S., Postel, E. A., Agarwal, A., Haines, J. L., Pericak-Vance, M. A., Rosenfeld, P. J. et al. (2004). Age-related maculopathy: a genomewide scan with continued evidence of susceptibility loci within the 1q31, 10q26, and 17q25 regions. The American Journal of Human Genetics *75*, 174–189.

Weeks, D. E. and Ott, J. (1990). Reply to Dr. Carothers: Support intervals for genetic risks. American journal of human genetics *47*, 166.

Welter, D., MacArthur, J., Morales, J., Burdett, T., Hall, P., Junkins, H., Klemm, A., Flicek, P., Manolio, T., Hindorff, L. et al. (2014). The NHGRI GWAS Catalog, a curated resource of SNP-trait associations. Nucleic acids research *42*, D1001–D1006.

Wray, N. R., Goddard, M. E. and Visscher, P. M. (2007). Prediction of individual genetic risk to disease from genome-wide association studies. Genome research *17*, 1520–1528.

Wu, J., Pfeiffer, R. M. and Gail, M. H. (2013). Strategies for Developing Prediction Models From Genome-Wide Association Studies. Genetic epidemiology *37*, 768–777.

Wu, T. T., Chen, Y. F., Hastie, T., Sobel, E. and Lange, K. (2009). Genome-wide association analysis by lasso penalized logistic regression. Bioinformatics *25*, 714–721.

Yang, J., Lee, S. H., Goddard, M. E. and Visscher, P. M. (2011). GCTA: a tool for genome-wide complex trait analysis. The American Journal of Human Genetics *88*, 76–82.

Zaidi, S., Choi, M., Wakimoto, H., Ma, L., Jiang, J., Overton, J. D., Romano-Adesman, A., Bjornson, R. D., Breitbart, R. E., Brown, K. K., Carriero, N. J., Cheung, Y. H., Deanfield, J., DePalma, S., Fakhro, K. A., Glessner, J., Hakonarson, H., Italia, M. J., Kaltman, J. R., Kaski, J., Kim, R., Kline, J. K., Lee, T., Leipzig, J., Lopez, A., Mane, S. M., Mitchell, L. E., Newburger, J. W., Parfenov, M., Pe'er, I., Porter, G., Roberts, A. E., Sachidanandam, R., Sanders, S. J., Seiden, H. S., State, M. W., Subramanian, S., Tikhonova, I. R., Wang, W., Warburton, D., White, P. S., Williams, I. A., Zhao, H., Seidman, J. G., Brueckner, M., Chung, W. K., Gelb, B. D., Goldmuntz, E., Seidman, C. E. and Lifton, R. P. (2013). De novo mutations in histone-modifying genes in congenital heart disease. Nature *498*, 220–3.