# MODEL-BASED CLASSIFICATION WITH APPLICATIONS TO HIGH-DIMENSIONAL DATA IN BIOINFORMATICS

A Dissertation

Presented to the Faculty of the Graduate School

of Cornell University

in Partial Fulfillment of the Requirements for the Degree of

Doctor of Philosophy

by

Muting Wan

January 2015

© 2015 Muting Wan
ALL RIGHTS RESERVED

MODEL-BASED CLASSIFICATION WITH APPLICATIONS TO HIGH-DIMENSIONAL DATA IN BIOINFORMATICS

Muting Wan, Ph.D.

Cornell University 2015

In recent years, sparse classification problems have emerged in many fields of study. Finite mixture models have been developed to facilitate Bayesian inference where parameter sparsity is substantial. Shrinkage estimation allows strength borrowing across features in light of the parallel nature of multiple hypothesis tests. Important examples that incorporate shrinkage estimation and finite mixture model for sparse classification include the hierarchical model in Smyth (2004) and the explicit mixture model in Bar et al. (2010) for Bayesian microarray analysis. Classification with finite mixture models is based on the posterior expectation of latent indicator variables. These quantities are typically estimated using the expectation-maximization (EM) algorithm in an empirical Bayes approach or Markov chain Monte Carlo (MCMC) in a fully Bayesian approach. MCMC is limited in applicability where high-dimensional data are involved because its sampling-based nature leads to slow computations and hard-to-monitor convergence. In a fully Bayesian framework, we investigate the feasibility and performance of variational Bayes (VB) approximation and apply the VB approach to fully Bayesian versions of several finite mixture models that have been proposed in bioinformatics. We find that it achieves desirable speed and accuracy in sparse classification with hierarchical mixture models for high-dimensional data.

Another example of sparse classification in bioinformatics solvable via

model-based approaches is expression quantitative trait loci (eQTL) detection, in which determining whether association between a gene and any given single nucleotide polymorphism (SNP) is significant is regarded as classifying genes as null or non-null with respect to the given SNP. High-dimensionality of the data not only causes difficulties in computations, but also renders the confounding impact of unwanted variation in the data irrefutable. Model-based approaches that account for unwanted variation by incorporating a factor analysis term representing hidden factors and their effects have been adopted in applications such as differential analysis and eQTL detection. HEFT (Gao et al., 2014) is a fast approach for model-based eQTL identification while simultaneously learning hidden effects. We develop a hierarchical mixture model-based empirical Bayes approach for sparse classification while simultaneously accounting for unwanted variation, as well as a family of model-based approaches that are its simplifications with the aim of attractive computational efficiency. We investigate feasibility and performance of these model-based approaches in comparison with HEFT using several real data examples in bioinformatics.

## BIOGRAPHICAL SKETCH

Muting Wan was born in Guangdong Province, China. In 2002, she moved abroad to study in the United Kingdom. She completed her A Levels studies at Notre Dame Sixth Form College in Leeds, United Kingdom in 2004. She then attended St Hugh's College, University of Oxford, United Kingdom and earned a Bachelor of Arts degree (Upper Second-Class Honours) in Mathematics and Statistics in 2007. Muting came to the United States in the same year, and received a Master of Arts degree in Statistics from Yale University, CT in 2008. In the fall of 2009, she began work on a Doctor of Philosophy degree in Statistics at Cornell University.

To my family

## ACKNOWLEDGEMENTS

I would like to express my appreciation to my advisors, Dr. James Booth and Dr. Martin Wells, for their wisdom, patience, advice, and support for this work. I would also like to thank Dr. Giles Hooker for serving on my committee. My appreciation also goes to Dr. Jason Mezey and Chuan Gao for providing the data used in this work.

I am thankful for financial support from Department of Statistical Science, Cornell University. I would like to thank Dr. Patrick Sullivan and Dr. Lynn Johnson for their help throughout my experience as their teaching assistant, and Francoise Vermeylen for her guidance in statistical consulting.

Several friends have been supportive throughout the various stages of this work, through discussions of technical issues or simply by sharing helpful perspective on the process of achieving a doctorate. In particular, I would like to acknowledge Cecilia Earls, James Li, Rajendran Narayanan, Darcy Steeg Morris, Luo Xiao, Irina Gaynanova, William Nicholson, and Jon Steingrimsson.

Most of all, I appreciate the constant support of my family, especially my father Haobo Wan, my mother Cuihua Liang, and my fiancé Chen Meng. I would like to thank brothers and sisters from the First Ithaca Chinese Christian Church for their love and prayers.

## TABLE OF CONTENTS


Biographical Sketch . . . iii
Dedication . . . iv
Acknowledgements . . . v
Table of Contents . . . vi
List of Tables . . . viii
List of Figures . . . x

**1 Introduction** **1**

**2 Hierarchical Mixture Model-Based Classification via Variational Bayes** **6**
2.1 Review of VB . . . 6
2.2 The VB Method . . . 9
2.2.1 Overview . . . 9
2.2.2 Practicality . . . 11
2.2.3 Over-confidence . . . 13
2.2.4 Simple two-component mixture model . . . 14
2.2.5 Marginal posterior approximation . . . 16
2.3 VB for a general finite mixture model . . . 19
2.3.1 Motivation . . . 19
2.3.2 The B-LIMMA model . . . 20

**3 Variational Bayes Numerical Illustrations** **24**
3.1 Simulation . . . 24
3.1.1 The B-LIMMA model . . . 24
3.1.2 A mixture model extended from the LIMMA model . . . 27
3.1.3 A mixture model for count data . . . 34
3.2 Real data examples . . . 38
3.2.1 APOA1 data . . . 39
3.2.2 Colon Cancer data . . . 39

**4 Model-Based Classification with Simultaneous Adjustment for Unwanted Variation** **43**
4.1 Extension of the B-LIMMA model . . . 43
4.2 Model-Based Classification with Simultaneous Adjustment for Unwanted Variation . . . 44
4.2.1 RRmix Model . . . 44
4.2.2 The family of models . . . 49
4.2.3 Practical considerations . . . 51
4.3 Connections with HEFT . . . 54
4.3.1 Connections between the family of models and HEFT . . . 54
4.3.2 Modifications from EM algorithm in HEFT . . . 56

**5 Model-Based Classification with Simultaneous Adjustment for Unwanted Variation Numerical Illustrations** **62**
5.1 Simulation Study . . . 62
5.1.1 Synthetic data . . . 62
5.1.2 Evaluation of models . . . 68
5.1.3 Identifiability . . . 74
5.1.4 Algorithmic issues . . . 78
5.1.5 Random or fixed error variance . . . 83
5.1.6 Random or fixed primary effect . . . 86
5.2 Applications . . . 89
5.2.1 eQTL analysis with Lung Airway Data . . . 89
5.2.2 Differential expression analysis of Microarray data with Alzheimer's Disease Data . . . 101
5.2.3 Count-based differential expression analysis of RNA-sequencing data with Zebrafish Data . . . 107

**6 Discussion** **110**

**A Technical Details of Chapter 3** **115**
A.1 VB-LEMMA algorithm . . . 115
A.1.1 The B-LEMMA model . . . 115
A.1.2 Algorithm . . . 116
A.2 VB-Proteomics algorithm . . . 121
A.2.1 The proteomics model . . . 121
A.2.2 Algorithm . . . 122

**B Technical Details of Chapter 4** **131**
B.1 RRmix model and EM algorithm . . . 131
B.1.1 E-step . . . 133
B.1.2 M-step . . . 140
B.1.3 Shrinkage estimation of gene-specific error variance . . . 141
B.1.4 Iterative Scheme . . . 142
B.2 FF model and EM algorithm . . . 145
B.2.1 E-step . . . 146
B.2.2 M-step . . . 146
B.2.3 Iterative Scheme . . . 147
B.2.4 Calculation of p-values . . . 147
B.3 FR model and EM algorithm . . . 149
B.4 RF model and EM algorithm . . . 151
B.5 RR model and EM algorithm . . . 155

**C Supplementary Information of Chapter 5** **158**

## LIST OF TABLES


3.1 Posterior estimates, deduced number of non-null genes, and TPR and accuracy for `lemma`, VB-LEMMA, and MCMC-LEMMA on simulated B-LEMMA data with $G = 5,000, n_{1g} \equiv 6, n_{2g} \equiv 8, \sigma_g^2 \sim IG(41, 400)$ (*i.i.d.*) and cutoff= 0.8. . . . . 30

3.2 Proteomics count data: simulation under the two models and VB-proteomics and MCMC-proteomics classifications. Total number of proteins is 1,307. Cutoff is 0.8. Accuracy is as defined in Section 3.1.1. . . . . 36

5.1 Number of detected associations that were produced by LM, HEFT, FR, and RRmix. For LM and FR, detections were made with a Bonferroni corrected threshold on the p-values. Non-duplicated detections produced by HEFT were extracted from Gao et al. (2014) among which we verified 61 are cis. For RRmix, detections were made with a threshold of 0.9999 on the posterior mean of the gene-specific latent indicator. . . . . 92

5.3 FR and HEFT p-values for each of 24 cis and non-duplicated associations that were found by FR but not by HEFT. Significant associations were determined by comparison to the Bonferroni corrected threshold $0.05/(7,575 \times 191,959) = 3.438578e\text{-}11$. HEFT p-values were extracted from data downloaded from the site at NCBI's Gene Expression Omnibus where authors of Gao et al. (2014) deposited their results. The last column shows references that indicate lung disease association of 9 genes. . . . . 94

5.4 Top SNP associations for genes GTF2H1, MTRR, TEFM, and RUVBL1 identified by HEFT and FR (right). . . . . 97

5.6 Cis and non-duplicated associations detected by our methods and LM but not by HEFT that involve genes that have previously been linked with lung disease. As cis and non-duplicated associations were considered, FR and RRmix identified different but nearby SNPs for gene NKIRAS1. . . . . 100

5.7 Number of X / Y genes found in the top 20 / 40 / 60 by RRmix, FR, HEFT-R, LM, and RUV-2 in the Alzheimer's study. All procedures were given unpreprocessed data as input. RRmix, FR, HEFT-R were implemented with 6 hidden factors and RUV-2 with 10 hidden factors. The highest number in each column is shown in bold. . . . . 106

C.1 List of top 100 non-duplicated eQTL associations identified by FR, with a Bonferroni corrected significance level $0.05/(7,575 \times 191,959)$. The columns from left to right are ranking, Ensemble code for the genes, SNP chromosome, SNP position, p-value of gene being non-null with respect to SNP, and TRUE/FALSE (T/F) for whether the gene-SNP association is cis. There are 64 cis associations in the list. . . . . 158
C.2 List of top 100 non-duplicated eQTL associations identified by RRmix, with a threshold of 0.9999 on the posterior non-null probability. The columns from left to right are ranking, Ensemble code for the genes, SNP chromosome, SNP position, posterior probability of gene being non-null with respect to SNP, and TRUE/FALSE (T/F) for whether the gene-SNP association is cis. There are 54 cis associations in the list. . . . . 160

## LIST OF FIGURES


2.1 Approximate marginal posteriors given by MCMC, VB and GBVA for simulated data under the simple two-component mixture model (2.1). In the last plot, the posterior mean of $b_g$, $g = 1, .., G$ is compared between VB and MCMC, and between GBVA and MCMC. . . . . 18

3.1 `limma` and VB-LIMMA classification comparison on simulated two-sample data with $G = 5,000, n_{1g} \equiv n_{2g} \equiv 8, \tau = 0, p \in \{0.05, 0.25\}, \nu = \frac{1}{2}, \sigma_g^2 \sim IG(41, 400)$ (*i.i.d.*). . . . . 25

3.2 `limma` and VB-LIMMA classification comparison on simulated two-sample data with $G = 5,000, n_{1g} \equiv n_{2g} \equiv 8, \tau = 0, p \in \{0.05, 0.25\}, \nu = 2, \sigma_g^2 \sim IG(41, 400)$ (*i.i.d.*). . . . . 26

3.3 Marginal posterior plots approximated by VB-LEMMA and by MCMC-LEMMA, for the parameters in the B-LEMMA model. For each parameter, the dotted and solid vertical lines represent the estimated posterior mean by VB and by MCMC respectively, and the dashed vertical line marks the estimated parameter value given by `lemma`. The bottom right plot shows the posterior mean of the gene-specific error variance $\sigma_g^2$ for each gene $g$, estimated by VB-LEMMA and by MCMC-LEMMA, and how this quantity varies across the G genes. The dotted line indicates the prior mean 10 of $\sigma_g^2$ since the prior distribution of $\sigma_g^2$ are i.i.d. $IG(41, 400)$. . . . . 31

3.4 Approximated densities on simulated data with $G = 5,000$. From left to right: mixture density approximated by VB-LEMMA, the three component densities approximated by VB-LEMMA, mixture density approximated by MCMC-LEMMA, and the three component densities approximated by MCMC-LEMMA. On the x-axis in the second and the fourth plots, inward ticks indicate the classified non-null 1 genes and outward ticks indicate the classified non-null 2 genes by the corresponding method. . . . . 32

3.5 Comparison between VB-LEMMA and MCMC-LEMMA with multiple simulation on an Intel Xeon L5410 2.33GHz, 8GB RAM computer. 30 data sets were simulated with same true parameter values in Table 3.1. MCMC-LEMMA results were post-processed so that results were corrected for label-switching among the simulated data sets. The dashed lines mark the true parameter values. 33

3.6 Simulated data under the Poisson GLMM (3.2), i.e. the "wrong" model. Top row: plots of log ratios of total counts in treatment and control groups against protein number with true non-null proteins, VB-proteomics labeled proteins, and MCMC-proteomics labeled proteins indicated. Bottom row: ROC curves, accuracy vs. cutoff, and FDR vs. cutoff for VB-proteomics, MCMC-proteomics, and for one protein at a time score tests. . . . 37
3.7 Simulated data under the Poisson-Gamma HGLM (3.3). Top row: plots of log ratios of total counts in treatment and control groups against protein number with true non-null proteins, VB-proteomics labeled proteins, and MCMC-proteomics labeled proteins indicated. Bottom row: ROC curves, accuracy vs. cutoff, and FDR vs. cutoff for VB-proteomics, MCMC-proteomics, and for one protein at a time score tests. . . . . 38
3.8 APOA1 data: histogram of $d_g$ with the eight non-null genes classified by VB-LIMMA indicated by inward ticks on the x-axis, and posterior mean of $b_g$ (i.e. posterior non-null probability of gene $g$) estimated by VB-LIMMA with the eight non-null gene numbers identified. . . . . 40
3.9 `limma` output: Top 8 genes detected as non-null for APOA1 data. Gene IDs are shown in the first column. . . . . 40
3.10 From left to right: Histogram of $d_g$ for Colon Cancer data, component densities estimated by VB-LEMMA, and component densities estimated by MCMC-LEMMA. On the x-axis, inward ticks indicate the classified non-null 1 genes and outward ticks indicate the classified non-null 2 genes with a 0.9 cutoff. . . . . 41

5.1 Distributions of between-gene correlations of $Y_{per}$ and $Y_{sim}$ in one of the large-sample simulated data and $Y_{dat}$ (left to right). . . . . 67
5.2 Comparison of t-statistic values produced by HEFT (Gao et al., 2014) (y-axis) to those by HEFT-R (x-axis) for the sample data distributed along with the HEFT software. The sample data contains 200 gene expression variables on 116 subjects, and 200 SNPs on the subjects. Number of hidden factors $q = 5$ was assumed for implementation of both methods. Most points and those at the largest magnitudes fall on the green dashed line of $y = x$. . . . . 70

5.3 Boxplot of area under ROC curve for false positive rate cutoff of 0.05 (top row) and 0.001 (bottom row) for each model implemented on gene expression data with (columns 1 and 3) and without added hidden effects (columns 2 and 4) in the large-sample (columns 1 and 2) and small-sample (columns 3 and 4) cases. The average number of true gene-SNP associations in the ten simulated data sets is 25 in the large-sample case, and 27 in the small-sample case. . . . . 71
5.4 Boxplot of false discovery rate (top row) and power (bottom row) boxplot for each model implemented on gene expression data with (columns 1 and 3) and without added hidden effects (columns 2 and 4) in the large-sample (columns 1 and 2) and small-sample (columns 3 and 4) cases. The average number of true gene-SNP associations in the ten simulated data sets is 25 in the large-sample case, and 27 in the small-sample case. . . . . 72
5.5 Boxplot of implementation time in minutes for each model implemented on gene expression data with (columns 1 and 3) and without added hidden effects (columns 2 and 4) in the large-sample (columns 1 and 2) and small-sample (columns 3 and 4) cases. The average number of true gene-SNP associations in the ten simulated data sets is 25 in the large-sample case, and 27 in the small-sample case. . . . . 73
5.6 Plot of $Q$ function evaluated at current parameter estimates against iteration number corresponding to implementation of the EM algorithm of RR for each of the 1,000 SNPs in each of the 10 simulated data sets in the large-sample with added hidden effects configuration. For simulated data sets 5, 7, and 10, the $Q$ function versus Iteration for offending SNPs are labeled in red crosses, whereas all others are labeled as black filled circles. Minimum number of iterations and offending SNP IDs, if any, are shown in the bottom-right legend. In each grid that corresponds to a simulated data set, there is significant overlap of the black filled circles due to the fact that convergence of the EM algorithm for 1,000 SNPs are plotted. . . . . 80

5.7 Empirical distribution of of $\beta_{1g}$, the SNP effect, for $g = 1, \ldots, G$ based on the estimates produced by the EM algorithm of RR, for SNPs for which non-null genes are detected in simulated data set 5 in the large-sample with added hidden effects case. Estimated density curve is added in blue. Red tick marks on the x-axis show the values of $\beta_{1g}$ for detected genes with respect to the given SNP. Number of detected hits by RR for each SNP is shown in the histogram title, along with the SNP ID. Total number of detected gene-SNP associations for the simulated data set is 127, majority of which is contributed by the offending SNPs 479 and 722. . . . . 81
5.8 Comparison of $m_g$ computed on original response data and initial shrinkage estimate of $\sigma_g^2$ used as initial values for $\sigma_g^2$ for all models (left), comparison of estimate of $\sigma_g^2$ produced by FF and that by FR (middle), and comparison of estimate of $\sigma_g^2$ produced by RF and that by RR (right), for analysis on SNP 1 of simulated data set 1 in the *small-sample* with added hidden effects case. Values of $m_g$ and estimates of $\sigma_g^2$ are plotted on the y-axis, and only the first 100 genes are plotted for less condensed illustrations. Each line corresponds to one gene. Effect of shrinkage is reflected by the shrunken range of line end points on the right side of each plot. . . . . 84
5.9 Comparison of $m_g$ computed on original response data and initial shrinkage estimate of $\sigma_g^2$ used as initial values for $\sigma_g^2$ for all models (left), comparison of estimate of $\sigma_g^2$ produced by FF and that by FR (middle), and comparison of estimate of $\sigma_g^2$ produced by RF and that by RR (right), for analysis on SNP 1 of simulated data set 1 in the *large-sample* with added hidden effects case. Values of $m_g$ and estimates of $\sigma_g^2$ are plotted on the y-axis, and only the first 100 genes are plotted for less condensed illustrations. Each line corresponds to one gene. Effect of shrinkage is reflected by the shrunken range of line end points on the right side of each plot. . . . . 85
5.10 Manhattan and QQ plots for the two genes for which a significant cis association was identified by LM but not by HEFT. A reference line at $-\log_{10}(0.05/(7,575 \times 191,959)) = 10.46$ for the Bonforroni threshold is indicated in the Manhattan plots. . . . . 92
5.11 Manhattan and QQ plots for the top two genes for which a significant cis association was identified by FR but not by HEFT. A reference line at $-\log_{10}(0.05/(7,575 \times 191,959)) = 10.46$ for the Bonforroni threshold is indicated in the Manhattan plots. . . . . 95

5.12 Manhattan and QQ plots on p-values computed via LM (red/orange) and FR (blue) for genes GTF2H1 and MTRR. A reference line at $-\log_{10}(0.05/(7,575 \times 191,959)) = 10.46$ for the Bonforroni threshold is indicated in the Manhattan plots. . . . . 97
5.13 Manhattan and QQ plots on p-values computed via LM (red/orange) and FR (blue) for genes TEFM and RUVBL1. A reference line at $-\log_{10}(0.05/(7,575 \times 191,959)) = 10.46$ for the Bonforroni threshold is indicated in the Manhattan plots. . . . . 98
5.14 A heat map of the subset of p-values that correspond to the first 10,000 SNPs (x-axis) and all 7,575 genes (y-axis) computed by LM (top), HEFT (bottom-left), and FR (bottom-right). The darker red a pixel, the smaller the p-value. Vertical lines are visible in the heat map for LM but not in the heat map for HEFT or FR, indicating successful removal of global impact of hidden factors while identifying significant associations by HEFT and by FR. . . 102
5.15 Plot of loading 1 and loading 2 (top), and of loading 2 and loading 3 (bottom), estimated by FR and by RRmix, when the models were implemented on the lung SAE data: all 7,575 genes and SNP rs1886498, the first SNP in the 96 non-duplicated associations reported in Gao et al. (2014). Separation of smokers (blue) and non-smokers (red) is clear in all plots, indicating correct recovery of smoking effect by FR and by RRmix when smoking status was treated as hidden. . . . . 103
5.16 Scree plot of all PCs (left) and the second to the twentieth PCs (right) based on principal component analysis on the gene expression correlation matrix of the Alzheimer's disease data. . . . 105
5.17 ROC curves for RRmix (solid), FR (dashed), and HEFT-R (dotted) with the 475 differentially expressed genes detected by RUVg as a gold standard. . . . . 109
5.18 Plot of the two hidden factor loadings estimated by RRmix, FR, and HEFT-R (left to right). No clustering by treatment is observed, which indicates little "wanted" variation was incorrectly captured in the estimated hidden factors by the methods, and hence is desirable. . . . . 109

C.1 Manhattan and QQ plots for the third and fourth (TPP2) genes for which a significant cis association is identified by FR but not by HEFT. Gene ranking is based on ascending p-value among the 24 extra cis associations. A reference line at $-\log_{10}(0.05/(7,575 \times 191,959)) = 10.46$ for the Bonforroni threshold is indicated in the Manhattan plots. Deviation from the reference line in the QQ plot and lack of unique peak in the Manhattan plots suggest that the cis association detected by FR for gene UBA1 (top) with gene ID 7317 is likely a false positive. . . . . 163

C.2 Manhattan and QQ plots for the fifth (SPINK5) and sixth genes for which a significant cis association is identified by FR but not by HEFT. Gene ranking is based on ascending p-value among the 24 extra cis associations. A reference line at $-\log_{10}$ of the Bonforroni threshold is indicated in the Manhattan plots. . . . . 164
C.3 Manhattan and QQ plots for the seventh and eighth (NKIRAS1) genes for which a significant cis association is identified by FR but not by HEFT. Gene ranking is based on ascending p-value among the 24 extra cis associations. A reference line at $-\log_{10}$ of the Bonforroni threshold is indicated in the Manhattan plots. . . 165
C.4 Manhattan and QQ plots for the ninth (BTG3) and tenth genes for which a significant cis association is identified by FR but not by HEFT. Gene ranking is based on ascending p-value among the 24 extra cis associations. A reference line at $-\log_{10}$ of the Bonforroni threshold is indicated in the Manhattan plots. . . . . 166
C.5 Manhattan and QQ plots for the eleventh and twelfth (MAN1C1) genes for which a significant cis association is identified by FR but not by HEFT. Gene ranking is based on ascending p-value among the 24 extra cis associations. A reference line at $-\log_{10}$ of the Bonforroni threshold is indicated in the Manhattan plots. . . 167
C.6 Manhattan and QQ plots for the fifteenth and sixteenth (YBEY) genes for which a significant cis association is identified by FR but not by HEFT. Gene ranking is based on ascending p-value among the 24 extra cis associations. A reference line at $-\log_{10}$ of the Bonforroni threshold is indicated in the Manhattan plots. . . 168
C.7 Manhattan and QQ plots for the nineteenth and twentieth (CCDC15) genes for which a significant cis association is identified by FR but not by HEFT. Gene ranking is based on ascending p-value among the 24 extra cis associations. A reference line at $-\log_{10}$ of the Bonforroni threshold is indicated in the Manhattan plots. . . . . 169
C.8 Manhattan and QQ plots for the twenty-first (RGS17) and twenty-second genes for which a significant cis association is identified by FR but not by HEFT. Gene ranking is based on ascending p-value among the 24 extra cis associations. A reference line at $-\log_{10}$ of the Bonforroni threshold is indicated in the Manhattan plots. . . . . 170

CHAPTER 1

# INTRODUCTION

Many problems in bioinformatics concern the identification of sparse non-null features in a high-dimensional and noisy null features space. The total number of features is typically large, often $10^4$ or more. Statistical inference therefore involves conducting one or more hypothesis tests for each of thousands of features. Family-wise error rate (FWER) procedures such as the Bonferroni correction or false discovery rate (FDR) control are usually adopted to control for multiple testing. The assumption of heterogeneous error variance across features is considered realistic. However, in making inference about each feature individually, power is reduced due to correlations among features. Moreover, t- and F-test statistics can be unstable, since the denominator of these test statistics involve an error variance estimate that is sensitive to small sample size, and small sample size is common in many bioinformatics data such as microarray data due to cost constraints. As reviewed in Xu (2013) and Bar et al. (2010), several proposed solutions modify the denominator of the test statistic, some of which are parametric approaches that are based on linear models. Among them, the moderated t-statistic proposed in Smyth (2004) arises from a hierarchical mixture model proposed by Lönnstedt and Speed (2002). In the inferential approach of Smyth (2004), instead of separate estimation of each feature-specific error variance, shrinking them towards a common value results in strength-borrowing across features. Bar et al. (2010) proposed an empirical Bayes estimate of the hyperparameters in the prior distribution of error variance that further strengthens shrinkage estimation of the feature-specific error variance in Smyth (2004).

Besides the simple hierarchical parametric model in Lönnstedt and Speed (2002) that underpins the hierarchical mixture model in Smyth (2004) and in turn Bar et al. (2010) in microarray data analysis, finite mixture models have been widely used for model-based classification in bioinformatics (McLachlan and Peel, 2004; Zhang et al., 2005; Efron, 2008). A finite mixture prior distribution is used in the model for Bayesian classification of quantitative trait loci (QTL) in Zhang et al. (2005). The two-groups model in Efron (2008) is a finite mixture model with an implicit latent indicator variable for detection of non-null genes from a vast number of null genes in microarray analysis. Mixtures of distributions provide flexibility in capturing complex data distributions that cannot be well described by a single standard distribution. Finite mixture models, most commonly finite Gaussian mixture models, facilitate classification where each component distribution characterizes a class of "similar" data points. Identifying non-null features among null features in a two-groups scenario naturally leads to sparse classification with a finite mixture of "null" and "non-null" component distributions, with component membership indicated by a latent indicator variable.

The empirical Bayes method used in Bar et al. (2010) is data-driven in that the hyperparameters are estimated from the marginal likelihood of the mean squared error for each feature. Hyperparameter estimates are then plugged into the posterior mean or mode of error variance which is a shrinkage estimate. Thus, a computational scheme that iteratively updates shrinkage estimates of the error variances and the other parameters in the model is called for. Treating the latent indicator in the hierarchical mixture model as missing data, an EM algorithm (McLachlan and Krishnan, 2007) is implemented to produce maximum a posteriori (MAP) estimates. The posterior mean of the latent indicator is then

utilized for classification. The empirical Bayes local false discovery rate estimate developed from the two-groups model in Efron (2008) is another important example of a finite mixture model.

Hierarchical mixture model-based classification addresses sparsity by allowing information sharing across features via shrinkage estimation in high-dimensional problems, in particular "large $p$ small $n$" problems where $p$, the number of features to be classified, is several orders of magnitude greater than $n$, the sample size. Typically, Bayesian classification results are obtained based on posterior expectation of the latent indicators (Bar et al., 2010) in an empirical Bayes approach. Alternatively, fully Bayesian inference based on the hierarchical structure of the model can be conducted via MCMC which approximates marginal posterior distributions, from which the posterior expectation of the latent indicators are calculated for classification. However, in high-dimensional problems it is difficult to assess convergence of fully Bayesian methods implemented using MCMC algorithms and the computational burden may be prohibitive. Fully Bayesian inference via MCMC procedures are therefore limited in practicality. Variational Bayes (VB) has been promoted as a fast deterministic method of approximating marginal posterior distributions, and therefore as an alternative to MCMC methods for Bayesian inference (Beal, 2003; Bishop, 2006; Ormerod and Wand, 2010). In the first part of this dissertation, we investigate the feasibility and performance of VB for hierarchical mixture models in sparse classification in a fully Bayesian framework.

High-dimensional data involved in bioinformatics applications are affected by noise. While the latent indicators in a hierarchical mixture model can guide identification of variation of interest and error variances, unwanted variation in

high-dimensional data can degrade inferential accuracy of naive model-based classification approaches. A major type of high-dimensional data that are susceptible to unwanted variation are expression data in application areas such as eQTL, microarray, RNA-Sequencing, and mass spectrometry data analysis which give rise to high-dimensional classification problems. Unwanted variation is generally recognized as the existence of hidden factors and their effects. The hidden factors are presumed to be unknown, unobserved, and confounding sources of variation. Hidden factors are particularly pronounced in gene expression data (Leek and Storey, 2007). Researchers generally employ linear models to estimate biological factors of interest in expression data while accounting for covariate effects. Failing to account for hidden factors may lead to decrease in power and an increase in false detections. Unwanted variation that confounds the biological variation of interest such as batch effects and other physical variables that correlate with batch effects (Gagnon-Bartsch and Speed, 2012) may be a pitfall in statistical inference.

Model-based classification methods typically include a term in the model that accounts for hidden effects and is computed via factor analysis or similar techniques (Price et al., 2006; Leek and Storey, 2007; Friguet et al., 2009; Listgarten et al., 2010; Stegle et al., 2010; Sun et al., 2012; Gagnon-Bartsch and Speed, 2012; Gao et al., 2014). Among these methods, either the hidden factors (and their effects) are simultaneously learned with the primary effect associated with the biological factor of interest, or the separately learned hidden factors are incorporated into the model among known covariates prior to estimation of the primary effect. Both these approaches aim to improve the estimation accuracy of the primary effect through adjusting for hidden factors. Several methods, as reviewed in Jacob et al. (2012), identify unwanted variation factors via singular

value decomposition (SVD). A commonly used approach for fitting factor analysis models is the EM algorithm (Bishop, 2006; McLachlan and Krishnan, 2007). It has been employed in model-based approaches to solve classification problems in bioinformatics with the factors being treated as missing data (Friguet et al., 2009; Gagnon-Bartsch and Speed, 2012). We extend the explicit mixture model in Bar et al. (2010) to address the presence of unwanted variation by including a factor analysis component that can be simultaneously learned with the primary effect. Our empirical Bayes approach is implemented via EM algorithm which produces either maximum likelihood (ML) estimate or MAP estimate for model parameters, and has a connection with HEFT (Gao et al., 2014). For enhanced computational efficiency, we consider special cases of our hierarchical mixture model for fast and accurate model-based classification in bioinformatics applications.

CHAPTER 2

# HIERARCHICAL MIXTURE MODEL-BASED CLASSIFICATION VIA VARIATIONAL BAYES

## 2.1 Review of VB

Variational Bayes (VB) methods in statistics arose from the family of variational approximation methods (Jaakkola, 2000) for performing approximate Bayesian inference for graphical models with latent variables (Bishop, 1999; Attias, 2000; Beal, 2003). Since then, VB has been promoted and employed in several fields of modern applications, such as signal processing (Smídl and Quinn, 2005; Blei and Jordan, 2006; Tzikas et al., 2008), political science (Grimmer, 2011), bioinformatics (Logsdon et al., 2010; Li and Sillanpää, 2012), and in medical research such as brain imaging (Friston et al., 2011; Goldsmith et al., 2011). In the early years, VB was applied to Gaussian mixture models, factor analysis, principal component analysis, hidden Markov models, and their mixtures, for model learning (Bishop, 1999; Ghahramani and Beal, 2000; Bishop et al., 2002; Beal, 2003) and model selection (Corduneanu and Bishop, 2001; Teschendorff et al., 2005; McGrory and Titterington, 2007). Since then, VB has also been used for learning nonlinear latent variable models (Honkela and Valpola, 2005; Salter-Townshend and Murphy, 2009), conducting functional regression analysis (Goldsmith et al., 2011), dealing with missing data in regression (Faes et al., 2011), and fitting location-scale models that contain elaborate distributional forms (Wand et al., 2011). Recently, Logsdon et al. (2010) applies VB based on the model proposed by Zhang et al. (2005) and shows that their VB solution outperforms single-marker testing in QTL analysis. Li et al. (2011) utilizes VB as an alternative to

Markov chain Monte Carlo (MCMC) for hierarchical shrinkage-based regression models in QTL mapping with epistasis.

VB has been promoted as a fast deterministic method of approximating marginal posterior distributions and therefore as an alternative to MCMC methods for Bayesian inference (Beal, 2003; Bishop, 2006; Ormerod and Wand, 2010). Posterior means computed from VB approximated marginal posterior density have been observed to be extremely accurate. For example, Bishop (2006) illustrates the VB solution of a simple hierarchical Gaussian model. Comparison with Gelman et al. (2003), where the true marginal posteriors for the same model is provided, shows that the VB solution in Bishop (2006) recovers the posterior mean exactly. However, it has also been observed that VB often underestimates posterior variance. This property is explained in Bishop (2006) to be due to the form of Kullback-Leibler divergence employed in the VB theory. In the setting where $n$ observations follow a Gaussian distribution with large known variance and a zero-mean Gaussian-distributed mean whose precision is assigned a Gamma prior, Rue et al. (2009) uses a simple latent Gaussian model to illustrate that VB may underestimate posterior variance of the precision parameter by a ratio of $O(n)$. Ormerod (2011) proposes a grid-based variational approximation (GBVA) method that corrects the variance in VB approximated marginal posterior densities for the same model.

In the context of finite mixture models, model complexity prevents exact marginal posteriors from being derived explicitly. For example, consider the following simple two-component mixture model in a fully Bayesian framework,

with observed data $\{d_g\}$ and known hyperparameters $\tau_0, \sigma^2_{\tau_0}, \psi_0, \sigma^2_{\psi_0}, a_0, b_0, \alpha_1, \alpha_0$:

$$d_g|b_g, \tau, \psi, \sigma^2 = (1 - b_g)N(\tau, \sigma^2) + b_g N(\tau + \psi, \sigma^2), \tag{2.1}$$

$$b_g|p \sim Bernoulli(p), \qquad \tau \sim N(\tau_0, \sigma^2_{\tau_0}), \qquad \psi \sim N(\psi_0, \sigma^2_{\psi_0}),$$

$$\sigma^2 \sim IG(a_0, b_0), \qquad p \sim Beta(\alpha_1, \alpha_0),$$

where $IG(a, b)$ denotes an Inverse-Gamma distribution with shape $a$ and scale $b$. Classification is conducted based on the magnitude of the posterior mean of the latent indicator $b_g$; a posterior mean close to 1 indicating membership in the non-null group for feature $g$. The integrated nested Laplace approximation (INLA) framework is another popular approximate Bayesian method, introduced in Rue et al. (2009), whose implementation is available in the `R-INLA` package (Martino and Rue, 2009). However, INLA does not fit any of the mixture models mentioned above because they are not members of the class of latent Gaussian models. In the context of model (2.1), numerical experiments show that MCMC produces believable approximate marginal posterior densities which can be regarded as proxies of the true marginal posteriors despite the label-switching phenomenon (Marin and Robert, 2007) well-known for finite mixture models. More general finite mixture models are common for (empirical and fully) Bayesian inference in high-dimensional application areas, such as bioinformatics. Computations for the two-groups model in Smyth (2004), for example, are burdensome due to presence of the gene-specific error variance and different variances for the two component distributions. Fully Bayesian inference via MCMC procedures are limited in practicality due to heavy computational burden resulting from the high-dimensionality of the data. We show that VB approximated densities are comparable to MCMC approximated ones in terms of both posterior mean and variance in this context, but VB achieves substantial gains in computational speed over MCMC.

## 2.2 The VB Method

### 2.2.1 Overview

VB is a deterministic estimation methodology based on a factorization assumption on the approximate joint posterior distribution. This is a free-form approximation in that implementation of VB does not start with any assumed parametric form of the posterior distributions. The free-form factors are approximate marginal posterior distributions that one tries to obtain according to minimization of the Kullback-Leibler divergence between the approximate joint posterior and the true joint posterior. As a tractable solution to the minimization problem, the optimal factors depend on each other, which naturally leads to an iterative scheme that cycles through an update on each factor. The convergence of the algorithm is monitored via a single scalar criterion. Ormerod and Wand (2010) and Chapter 10 of Bishop (2006) present detailed introduction to VB in statistical terms. The VB approximation in this article refers to variational approximation to the joint and marginal posterior distributions under, in terminology used in Ormerod and Wand (2010), the product density (independence) restriction. The concept of a mean-field approximation has its roots in statistical physics and is a form of VB approximation under a more stringent product density restriction that the approximate joint posterior fully factorizes. The basic VB theory is outlined as follows.

Consider a Bayesian model with observed data $\mathbf{y}$, latent variables $\mathbf{x}$ and parameters $\mathbf{\Theta} = \{\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, ..., \boldsymbol{\theta}_m\}$. Denote $H = \{\mathbf{x}, \mathbf{\Theta}\}$ as the collection of all unknown quantities. The posterior density $p(H|\mathbf{y}) = p(\mathbf{y}, H)/p(\mathbf{y})$ is not necessarily tractable due to the integral involved in computing the marginal data density

$p(\mathbf{y})$. This intractability prevents computing marginal posterior densities such as $p(\boldsymbol{\theta}_i|\mathbf{y}) = \int p(H|\mathbf{y})d\{-\boldsymbol{\theta}_i\}$ where $\{-\boldsymbol{\theta}_i\}$ refers to $H\backslash\{\boldsymbol{\theta}_i\} = \{\mathbf{x}, \boldsymbol{\theta}_1, \boldsymbol{\theta}_2, ..., \boldsymbol{\theta}_{i-1}, \boldsymbol{\theta}_{i+1}, ..., \boldsymbol{\theta}_m\}$. Two key elements, namely minimization of the Kullback-Leibler divergence and employment of the product density restriction, underpin the VB algorithm. The Kullback-Leibler divergence gives rise to a scalar convergence criterion that governs the iterative algorithm. Through the product density restriction one assumes independence which allows derivation of tractable density functions.

For an arbitrary density for $H$, $q(H)$,

$$\log p(\mathbf{y}) = \int q(H) \log\left(\frac{p(\mathbf{y}, H)}{q(H)}\right) dH + \int q(H) \log\left(\frac{q(H)}{p(H|\mathbf{y})}\right) dH. \tag{2.2}$$

The second integral on the right hand side of (2.2) is the Kullback-Leibler divergence between $q(H)$ and $p(H|\mathbf{y})$, i.e. $D_{KL}(q(H)||p(H|\mathbf{y}))$ which is non negative. Hence,

$$\log p(\mathbf{y}) \geq \int q(H) \log\left(\frac{p(\mathbf{y}, H)}{q(H)}\right) dH = C_q(\mathbf{y}),$$

where $C_q(\mathbf{y})$ denotes the lower bound on the log of the marginal data density. This lower bound depends on the density $q$. Given a data set, $\log p(\mathbf{y})$ is a fixed quantity. This means that if a density $q$ is sought to minimize $D_{KL}(q(H)||p(H|\mathbf{y}))$, $q$ also maximizes $C_q(\mathbf{y})$ making the lower bound a viable approximation to $\log p(\mathbf{y})$.

The product density restriction states that $q(H)$ factorizes into some partition of $H$, i.e. $q(H) = \prod_{i=1}^{k} q_i(\mathbf{h}_i)$, but the parametric form of the $q_i(\mathbf{h}_i)$ factors is not specified. It can be shown that, under the product density restriction, the optimal $q_i(\mathbf{h}_i)$ takes the following form (Ormerod and Wand, 2010):

$$q_i(\mathbf{h}_i) \propto \exp\{E_{-\mathbf{h}_i}(\log p(\mathbf{y}, H))\} \tag{2.3}$$

for $i = 1, .., k$, where expectation is taken over all unknown quantities except $\mathbf{h}_i$ with respect to the density $\prod_{j\neq i}^{k} q_j(\mathbf{h}_j)$.

Since each variational posterior $q_i(\mathbf{h}_i)$ depends on other variational posterior quantities, the algorithm involves iteratively updating $q_i(\mathbf{h}_i), 1 \leq i \leq k$ until the increase in $C_q(\mathbf{y})$, computed in every iteration after each of the updates $q_i(\mathbf{h}_i), 1 \leq i \leq k$ has been made, is negligible. Upon convergence, approximate marginal posterior densities $q^*{}_i(\mathbf{h}_i), 1 \leq i \leq k$ are obtained, as well as an estimate $C_{q^*}(\mathbf{y})$ of the log marginal data density $\log p(\mathbf{y})$.

### 2.2.2 Practicality

As noted in Faes et al. (2011), under mild assumptions convergence of the VB algorithm is guaranteed (Luenberger and Ye, 2008, pg. 253). Bishop (2006) refers to Boyd and Vandenberghe (2004) and states that convergence is guaranteed because the lower bound $C_q(\mathbf{y})$ is convex with respect to each of the factors $q_i(\mathbf{h}_i)$. Convergence is easily monitored through the scalar quantity $C_q(\mathbf{y})$. Moreover, upon convergence, since $C_{q^*}(\mathbf{y})$ approximates $\log p(\mathbf{y})$, VB can be used to compare solutions produced under the same model but with different starting values or different orders of updating the $q$-densities. Because the optimal set of starting values corresponds to the largest $C_{q^*}(\mathbf{y})$ upon convergence, starting values can be determined empirically with multiple runs of the same VB algorithm. Experiments on starting values are generally undemanding due to high computational efficiency of the VB method, and are especially useful for problems where the true solutions are multimodal.

Computational convenience of VB is achieved if conjugate priors are assigned for model parameters and the complete-data distribution belongs to the exponential family. Beal (2003) describes models that satisfy these conditions

as conjugate-exponential models. In these models the approximate marginal posterior densities $q_i(\mathbf{h}_i)$ take the conjugate form and the VB algorithm only requires finding the characterizing parameters. The induced conjugacy plays an important role in producing analytical solutions. Feasibility of VB for models with insufficient conjugacy is an open research area.

In practice, it suffices to impose some relaxed factorization of $q(H)$ as long as the factorization allows derivation of a tractable solution. Although in many cases the chosen product density restriction leads to an induced product form of factors of $q(H)$, the imposed factorization and induced factorization of $q(H)$ are generally not equivalent and may lead to different posterior independence structures.

As an introductory example, consider a simple fully Bayesian model with a $N(\mu, \sigma^2)$ data distribution and a bivariate Normal-Inverse-Gamma prior $p_{(\mu,\sigma^2)}(\mu, \sigma^2) = p_{\mu|\sigma^2}(\mu|\sigma^2)p_{\sigma^2}(\sigma^2)$ with $p_{\mu|\sigma^2}(\mu|\sigma^2) = N(\mu_0, \lambda_0\sigma^2)$ and $p_{\sigma^2}(\sigma^2) = IG(A_0, B_0)$. Imposing the product density restriction $q_{(\mu,\sigma^2)}(\mu, \sigma^2) = q_\mu(\mu)q_{\sigma^2}(\sigma^2)$ leads to independent normal marginal posterior $q_\mu(\mu)$ and Inverse-Gamma marginal posterior $q_{\sigma^2}(\sigma^2)$. However, not imposing the product density restriction in this case leads to the bivariate Normal-Inverse-Gamma joint posterior $q_{\mu|\sigma^2}(\mu|\sigma^2)q_{\sigma^2}(\sigma^2)$ that reflects a different posterior independence structure. Hence, although choice of prior and product density restriction is problem-based, caution should be taken in examining possible correlation between model parameters, because imposing too much factorization than needed risks poor VB approximations if dependencies between the hidden quantities are necessary. This type of degradation in VB accuracy is noted in Ormerod and Wand (2010). There is a trade-off between tractability and accuracy with any type of

problem simplification via imposed restrictions.

### 2.2.3 Over-confidence

Underestimation of posterior variance of the VB approximate density has been observed in different settings such as those in Wang and Titterington (2005), Consonni and Marin (2007), and Rue et al. (2009). We review a possible reason why the variance of the approximate posterior given by VB tends to be smaller than that of the true posterior.

As explained previously by (2.2), a density $q(H)$ is sought to minimize the Kullback-Leibler divergence between itself and the true joint posterior $p(H|\mathbf{y})$. Upon examination of the form of the Kullback-Leibler divergence used in the VB method,

$$D_{KL}(q(H)||p(H|\mathbf{y})) = \int q(H)\left(-\log\left(\frac{p(H|\mathbf{y})}{q(H)}\right)\right)dH,$$

Bishop (2006) points out that, in regions of $H$ space where density $p(H|\mathbf{y})$ is close to zero, $q(H)$, the minimizer, has to be close to zero also; otherwise the negative log term in the integrand would result in a large positive contribution to the Kullback-Leibler divergence. Thus, the optimal $q^*(H)$ tends to avoid regions where $p(H|\mathbf{y})$ is small. The factorized form of $q^*(H)$ implies that each of the factor $q^*$-densities in turn tends to avoid intervals where the true marginal posterior density is small. Thus, the marginal posterior densities approximated by VB are likely to be associated with underestimated posterior variance and a more compact shape than the true marginal posterior densities.

### 2.2.4 Simple two-component mixture model

For model (2.1) where $g = 1, 2, ..., G$, with known hyperparameters $\tau_0, \sigma^2_{\tau_0}, \psi_0, \sigma^2_{\psi_0}$, $a_0, b_0, \alpha_1, \alpha_0$, observed data $E = \{d_g\}$ and hidden quantities $H = \{\tau, \psi, \sigma^2, p, \{b_g\}\}$, imposing the product density restriction $q(H) = q_{\{b_g\}}(\{b_g\}) \times q_{(\tau,p)}(\tau, p) \times q_\psi(\psi) \times q_{\sigma^2}(\sigma^2)$ results in $q$-densities $q_{\{b_g\}}(\{b_g\}) = \prod_g q_{b_g}(b_g) = \prod_g Bernoulli\left(\frac{\exp(\hat{\rho}_{1g})}{\exp(\hat{\rho}_{1g}) + \exp(\hat{\rho}_{0g})}\right)$, $q_\tau(\tau) = N(\widehat{M_\tau}, \widehat{V_\tau})$, $q_p(p) = Beta(\hat{\alpha}_1, \hat{\alpha}_0)$, $q_\psi(\psi) = N(\widehat{M_\psi}, \widehat{V_\psi})$, and $q_{\sigma^2}(\sigma^2) = IG(\hat{a}, \hat{b})$. Optimal $q$-densities are found by employing the following iterative scheme for computing variational parameters.

i) Set $\hat{a} = \frac{G}{2} + a_0$. Initialize $\widehat{M_{\sigma^{-2}}} = 1$, $\hat{\rho}_{0g} = \hat{\rho}_{1g} = 0$ for each $g$, and

$$\widehat{M_{b_g}} = \begin{cases} 1 & \text{if rank}(d_g) \geq 0.9G \\ 0 & \text{otherwise} \end{cases} \quad \text{for each } g,$$

$$\widehat{M_\psi} = \left| \sum_{g=1}^{G} d_g - \sum_{\{g:\text{rank}(d_g)\geq 0.9G\}} d_g \right|.$$

ii) Update

$$\widehat{M_\tau} \leftarrow \frac{\sum_g \widehat{M_{\sigma^{-2}}} \left( (1 - \widehat{M_{b_g}}) d_g + \widehat{M_{b_g}} (d_g - \widehat{M_\psi}) \right)}{G \widehat{M_{\sigma^{-2}}} + \frac{1}{\sigma^2_{\tau_0}}}$$

$$\widehat{M_\psi} \leftarrow \frac{\sum_g \widehat{M_{\sigma^{-2}}} \widehat{M_{b_g}} (d_g - \widehat{M_\tau})}{\sum_g \widehat{M_{b_g}} \widehat{M_{\sigma^{-2}}} + \frac{1}{\sigma^2_{\psi_0}}}$$

$$\hat{b} \leftarrow \frac{1}{2} \sum_g \left( \frac{1}{G \widehat{M_{\sigma^{-2}}} + \frac{1}{\sigma^2_{\tau_0}}} + \frac{\widehat{M_{b_g}}}{\sum_g \widehat{M_{b_g}} \widehat{M_{\sigma^{-2}}} + \frac{1}{\sigma^2_{\psi_0}}} \right)$$
$$+ \frac{1}{2} \sum_g \left( (1 - \widehat{M_{b_g}}) (d_g - \widehat{M_\tau})^2 + \widehat{M_{b_g}} (d_g - \widehat{M_\tau} - \widehat{M_\psi})^2 \right) + b_0$$

$$\widehat{M_{\sigma^{-2}}} \leftarrow \frac{\hat{a}}{\hat{b}}$$

$$\hat{\rho}_{1g} \leftarrow \frac{-\widehat{M_{\sigma^{-2}}}}{2}\left(\frac{1}{G\widehat{M_{\sigma^{-2}}}+\frac{1}{\sigma^2_{\tau_0}}}+\frac{1}{\sum_g \widehat{M_{b_g}}\widehat{M_{\sigma^{-2}}}+\frac{1}{\sigma^2_{\psi_0}}}+(d_g-\widehat{M_\tau}-\widehat{M_\psi})^2\right)+\widehat{\log p}$$

$$\hat{\rho}_{0g} \leftarrow \frac{-\widehat{M_{\sigma^{-2}}}}{2}\left(\frac{1}{G\widehat{M_{\sigma^{-2}}}+\frac{1}{\sigma^2_{\tau_0}}}+(d_g-\widehat{M_\tau})^2\right)+\widehat{\log(1-p)}$$

$$\widehat{M_{b_g}} \leftarrow \frac{\exp(\hat{\rho}_{1g})}{\exp(\hat{\rho}_{1g})+\exp(\hat{\rho}_{0g})}.$$

iii) Repeat ii) until the increase in

$$\begin{aligned}
C_q(\mathbf{y}) = \sum_g \Bigg\{ & -\frac{1}{2}\log(2\pi) - \frac{\widehat{M_{\sigma^{-2}}}}{2}\times(1-\widehat{M_{b_g}})\left(\frac{1}{G\widehat{M_{\sigma^{-2}}}+\frac{1}{\sigma^2_{\tau_0}}}+(d_g-\widehat{M_\tau})^2\right) \\
& - \frac{\widehat{M_{\sigma^{-2}}}}{2}\times\widehat{M_{b_g}}\left(\frac{1}{G\widehat{M_{\sigma^{-2}}}+\frac{1}{\sigma^2_{\tau_0}}}+\frac{1}{\sum_g \widehat{M_{b_g}}\widehat{M_{\sigma^{-2}}}+\frac{1}{\sigma^2_{\psi_0}}}+(d_g-\widehat{M_\tau}-\widehat{M_\psi})^2\right)\Bigg\} \\
& - \sum_g \left\{\widehat{M_{b_g}}\log\widehat{M_{b_g}} + (1-\widehat{M_{b_g}})\log(1-\widehat{M_{b_g}})\right\} \\
& + \log\left(Beta\left(\sum_g \widehat{M_{b_g}}+\alpha_1, \sum_g(1-\widehat{M_{b_g}})+\alpha_0\right)\right) - \log\left(Beta(\alpha_1,\alpha_0)\right) \\
& + \frac{1}{2}\left(\log\left(\frac{1}{G\widehat{M_{\sigma^{-2}}}+\frac{1}{\sigma^2_{\tau_0}}}\right) - \log\sigma^2_{\tau_0} + 1 - \frac{\frac{1}{G\widehat{M_{\sigma^{-2}}}+\frac{1}{\sigma^2_{\tau_0}}}+\left(\widehat{M_\tau}-\tau_0\right)^2}{\sigma^2_{\tau_0}}\right) \\
& + \frac{1}{2}\left(\log\left(\frac{1}{\sum_g \widehat{M_{b_g}}\widehat{M_{\sigma^{-2}}}+\frac{1}{\sigma^2_{\psi_0}}}\right) - \log\sigma^2_{\psi_0} + 1\right. \\
& \left. - \frac{\frac{1}{\sum_g \widehat{M_{b_g}}\widehat{M_{\sigma^{-2}}}+\frac{1}{\sigma^2_{\psi_0}}}+\left(\widehat{M_\psi}-\psi_0\right)^2}{\sigma^2_{\psi_0}}\right) \\
& + a_0\log b_0 - \log\Gamma(a_0) - \frac{b_0\hat{a}}{\hat{b}} - \hat{a}\log(\hat{b}) + \log\Gamma(\hat{a}) + \hat{a}
\end{aligned}$$

from previous iteration becomes negligible.

iv) Upon convergence, the remaining variational parameters are computed:

$$\widehat{V_\tau} \leftarrow \frac{1}{G\widehat{M_{\sigma^{-2}}} + \frac{1}{\sigma^2_{\tau_0}}} \qquad \text{and} \qquad \widehat{V_\psi} \leftarrow \frac{1}{\sum_g \widehat{M_{b_g}}\widehat{M_{\sigma^{-2}}} + \frac{1}{\sigma^2_{\psi_0}}}.$$

Here, we set starting values for $\widehat{M_{b_g}}$ such that $\widehat{M_{b_g}} = 1$ for genes that correspond to the top 10% of $d_g$ values and $\widehat{M_{b_g}} = 0$ otherwise. Classification with a different set of starting values such that $\widehat{M_{b_g}} = 1$ for genes that correspond to the top 5% and bottom 5% of $d_g$ values and $\widehat{M_{b_g}} = 0$ otherwise leads to almost the same results. In practice, we recommend using the "top-5%-bottom-5%" scheme of setting starting values for $\widehat{M_{b_g}}$ in the VB algorithm for classification with little prior information on location of the mixture component distributions rather than randomly generating values for $\widehat{M_{b_g}}$ - empirical evidence suggests that VB with randomly generated labels does not produce reasonable solutions.

### 2.2.5 Marginal posterior approximation

To investigate we simulated 20,000 values from the simple two-component model (2.1) with true values $\tau = 0$, $\psi = 20$, $p = 0.2$, $b_g \sim Bernoulli(p)$ *i.i.d.* for each $g$, and $\sigma^2 = 36$. In the Bayesian analysis $\tau$ and $\psi$ were assigned $N(0, 100)$ priors, $\sigma^2$ an $IG(0.1, 0.1)$ prior, and $p$ a $Beta(0.1, 0.9)$ prior. Starting values and priors used in VB, MCMC, and the base VB of GBVA are kept the same for comparison. MCMC was implemented in WinBUGS via R with 1 chain of length 20,000, a burn-in 15,000, and a thinning factor of 10. Figure 2.1 shows that VB-approximated marginal posterior densities closely match the MCMC-approximates. In particular, the posterior mean of $\{b_g\}$ estimated by VB, when plotted against posterior mean of $\{b_g\}$ estimated by MCMC, almost coincide

with the intercept zero slope one reference line, indicating little difference between VB and MCMC in terms of classification. In fact, with 3972 simulated non-null genes and a 0.8 cutoff for classification, MCMC identified 3303 genes as non-null with true positive rate 0.803 and false positive rate 0.00705, and VB also detected 3291 genes as non-null with true positive rate 0.800 and the same false positive rate as MCMC. Moreover, on an Intel Core i5-2430M 2.40GHz, 6GB RAM computer, it took VB 1.26 seconds to reach convergence with a $10^{-6}$ error tolerance, whereas MCMC procedure took about 13 minutes. Both methods are suitable for classification and posterior inference, with VB demonstrating an advantage in speed, in this example.

The grid-based variational approximation (GBVA) method detailed in Ormerod (2011) was developed to correct for over-confidence in VB densities. The method involves running a base VB algorithm and subsequent VB algorithms for density approximation with the aid of numerical interpolation and integration. For each marginal posterior density of interest, re-applying VB over a grid of values in GBVA is similar in spirit to re-applying Laplace approximation in the INLA Rue et al. (2009) method, and is the key step towards refining the marginal posterior density approximated by the base VB algorithm. We observe in Figure 2.1 that, although GBVA appears to correct for underestimation of posterior variance of VB, loss in accuracy occurs in estimation of posterior mean in comparison with MCMC. Experiments with grid selection and precision of numerical methods did not correct the shift in posterior mean, and a systematic error caused by grid-based calculations of the unnormalized density is suspected. Although the shifts in posterior mean of GBVA densities observed in Figure 2.1 are small in scale, for posterior inference the mismatch in mean may incur more serious loss in accuracy than underestimation of posterior vari-

ance of VB densities. Moreover, in the example, the total run time of GBVA (almost 6 hours) is even larger than that of MCMC (13 minutes). The undesirable speed of GBVA is due to the fact that refining the marginal posterior of the latent variable $\{b_g\}$ must go through two iterations, i.e. two grid points 0 and 1, for each one of the G genes. Therefore, in GBVA implementation, this step alone requires $2 \times G$ iterations. This example suggests that for a "large $p$ small $n$" problem, the approach of calculation over grid values can easily render the GBVA method computationally infeasible.

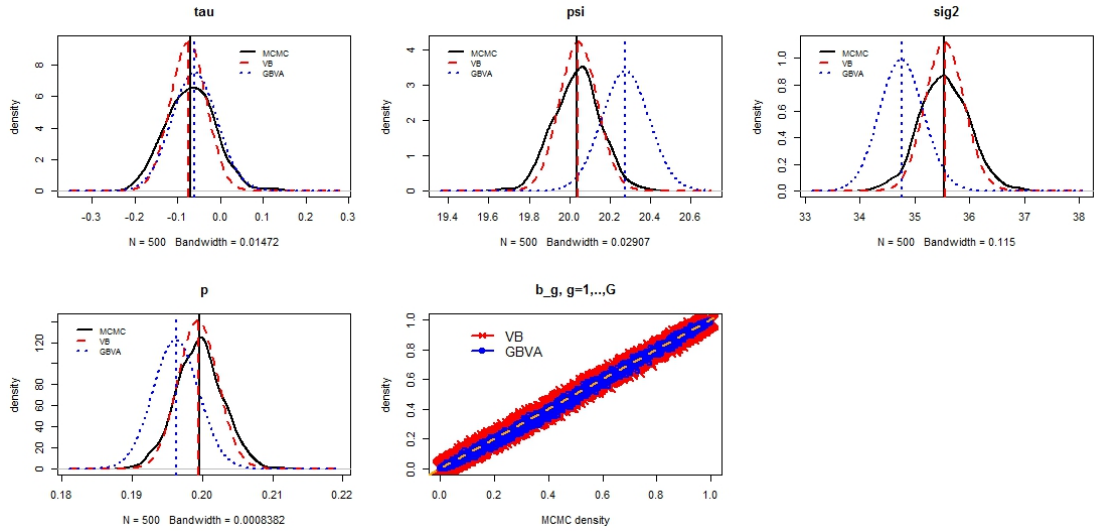


Figure 2.1: Approximate marginal posteriors given by MCMC, VB and GBVA for simulated data under the simple two-component mixture model (2.1). In the last plot, the posterior mean of $b_g$, $g = 1, .., G$ is compared between VB and MCMC, and between GBVA and MCMC.

## 2.3 VB for a general finite mixture model

### 2.3.1 Motivation

For large-scale problems that involve complex models on high-dimensional data, it is natural to consider MCMC for approximation of notoriously intractable marginal posterior densities. However, in many cases implementation of MCMC is impractical. MCMC, because of its sampling-based nature, becomes cumbersome for hierarchical "large $p$ small $n$" models where high-dimensional unknown quantities on various hierarchies are inter-dependent. To achieve desirable accuracy of MCMC therefore requires (multiple) long chains. Yet convergence of high-dimensional MCMC chains is difficult to monitor (Cowles and Carlin, 1996), and optimal MCMC chain settings often have to be determined based on heuristics. Moreover, in mixture models, an MCMC chain may produce evaluation of a marginal posterior density that is the same as other chains but associated with different component labels. The label-switching issue has been noted, for example in De Freitas et al. (2001), Marin and Robert (2007), and Grimmer (2011). Without correction of label-switching, misleading classification may arise in cases where multiple chains are used and the MCMC solution, taken as an average over chains with different or even contrasting labels, is wrong.

A common strategy for improving efficiency in approximation problems is to make simplifying assumptions so that plausible solutions for the complicated problem are obtained with satisfactory speed and accuracy. This is the motivation behind implementing VB for approximation of marginal posterior densities for finite mixture models in sparse classification problems. In this context, VB

is promising in achieving comparable accuracy to MCMC given limited computational resources, with easy-to-monitor convergence and no label-switching due to its deterministic nature, in scenarios where MCMC implementation is unattractive.

### 2.3.2 The B-LIMMA model

We illustrate a VB implementation for a general finite mixture models using a fully Bayesian version of the LIMMA model introduced by Smyth (2004).

The LIMMA model is designed for testing significance of differential gene expression in microarray experiments via individual moderated t-tests and posterior log odds ratios. Model parameters can be estimated via an empirical Bayes approach, from which closed form test-statistics can be obtained. For a two-sample experimental design, let $y_{ijg}$ denote the normalized log expression of gene $g$ in sample $j$ from treatment group $i$, where $g = 1, ..., G$, $i = 1, 2$, and $j = 1, ..., n_{ig}$. Then $d_g = \bar{y}_{2.g} - \bar{y}_{1.g}$ represents the observed differential expression for gene $g$, and $m_g = \sum_i \sum_j^{n_{ig}} (y_{ijg} - \bar{y}_{i.g})^2 / f_g$, where $f_g = n_{1g} + n_{2g} - 2$, is the mean squared error. A fully Bayesian version of the LIMMA model in this context is

the following hierarchical mixture model:

$$
\begin{aligned}
& d_g|b_g, \psi_g, \sigma^2_{\epsilon,g}, \tau = \tau + b_g\psi_g + \epsilon_g, \\
& m_g|\sigma^2_{\epsilon,g} \sim \frac{\sigma^2_{\epsilon,g}\chi^2_{f_g}}{f_g}, \quad \text{where } f_g := n_{1g} + n_{2g} - 2, \\
& \psi_g|\nu, \sigma^2_{\epsilon,g} \sim N(0, \nu\sigma^2_{\epsilon,g}), \\
& \epsilon_g|\sigma^2_{\epsilon,g} \sim N(0, \sigma^2_g), \quad \text{where } \sigma^2_g := \sigma^2_{\epsilon,g}c_g, \quad c_g := \frac{1}{n_{1g}} + \frac{1}{n_{2g}}, \\
& b_g|p \sim \textit{Bernoulli}(p) \quad \textit{i.i.d.}, \qquad \sigma^2_{\epsilon,g} \sim \textit{IG}\left(A_\epsilon, B_\epsilon\right) \ \textit{i.i.d.}, \\
& \tau \sim N\left(\mu_{\tau_0}, \sigma^2_{\tau_0}\right), \qquad \nu \sim \textit{IG}\left(A_\nu, B_\nu\right), \qquad p \sim \textit{Bernoulli}\left(\alpha_1, \alpha_0\right).
\end{aligned}
\tag{2.4}
$$

In this model $\tau$ represents the overall mean treatment difference, $\psi_g$ is the gene-specific effect of the treatment, and $b_g$ is a latent indicator which takes the value 1 if gene $g$ is non-null, i.e. is differentially expressed in the two treatment groups.

We refer to the above model as the B-LIMMA model. Derivation of the VB algorithm for the B-LIMMA model, which we refer to as VB-LIMMA, is as follows.

The set of observed data and the set of unobserved data are identified as $\mathbf{y}$ and $H$, respectively. For the B-LIMMA model, $\mathbf{y} = \{\{d_g\}, \{m_g\}\}$ and $H = \{\{b_g\}, \{\psi_g\}, \{\sigma^2_g\}, \tau, \nu, p\}$. The complete data log likelihood is

$$
\begin{aligned}
\log p(\mathbf{y}, H) = & \sum_g \log p(d_g|b_g, \psi_g, \sigma^2_g, \tau) + \sum_g \log p(m_g|\sigma^2_g) + \sum_g \log p(\psi_g|\nu, \sigma^2_g) \\
& + \sum_g \log p(b_g|p) + \log p(\tau) + \log p(\nu) + \sum_g \log p(\sigma^2_g) + \log p(p).
\end{aligned}
$$

In particular, the observed data log likelihood involves

$$\log p(d_g|b_g,\psi_g,\sigma_g^2,\tau) = \frac{1}{2}\log\left(2\pi\sigma_g^{-2}\right) - \frac{1}{2\sigma_g^2}\left[(1-b_g)(d_g-\tau)^2 + b_g(d_g-\tau-\psi_g)^2\right],$$

$$\log p(m_g|\sigma_g^2) = -\log\left(\frac{\sigma_g^2}{c_g f_g}\right) - \log\left(2^{\frac{f_g}{2}}\Gamma\left(\frac{f_g}{2}\right)\right) + \left(\frac{f_g}{2}-1\right)\log\left(\frac{m_g c_g f_g}{\sigma_g^2}\right) - \frac{1}{2}\left(\frac{m_g c_g f_g}{\sigma_g^2}\right).$$

The product density restriction is imposed such that $q(H) = q_{\{b_g\}}(\{b_g\}) \times q_{\{\psi_g\}}(\{\psi_g\}) \times q_{\{\sigma_g^2\}}(\{\sigma_g^2\}) \times q_{(\tau,\nu,p)}(\tau,\nu,p)$. Further factorizations are induced by applying (2.3):

$$q_{\{b_g\}}(\{b_g\}) = \prod_g q_{b_g}(b_g), \qquad q_{\{\psi_g\}}(\{\psi_g\}) = \prod_g q_{\psi_g}(\psi_g),$$

$$q_{\{\sigma_g^2\}}(\{\sigma_g^2\}) = \prod_g q_{\sigma_g^2}(\sigma_g^2), \quad \text{and} \quad q_{(\tau,\nu,p)}(\tau,\nu,p) = q_\tau(\tau)q_\nu(\nu)q_p(p).$$

Then, approximate marginal posterior distributions, with $\widehat{M_\theta}$ denoting the variational posterior mean $\int \theta q_\theta(\theta)d\theta$, and $\widehat{V_\theta}$ denoting the variational posterior variance $\int(\theta-\widehat{M_\theta})^2 q_\theta(\theta)d\theta$, are $q_\tau(\tau) = N(\widehat{M_\tau},\widehat{V_\tau})$, $q_\nu(\nu) = IG(\widehat{A_\nu},\widehat{B_\nu})$, $q_p(p) = Beta(\widehat{\alpha_1},\widehat{\alpha_0})$, $q_{\psi_g}(\psi_g) = N(\widehat{M_{\psi_g}},\widehat{V_{\psi_g}})$, $q_{\sigma_g^2}(\sigma_g^2) = IG(\widehat{A_{\sigma_g^2}},\widehat{B_{\sigma_g^2}})$, and $q_{b_g}(b_g) = Bernoulli(\widehat{M_{b_g}})$. Updating the $q$-densities in an iterative scheme boils down to updating the variational parameters in the scheme. Convergence is monitored via the scalar quantity $C_q(\mathbf{y})$, the lower bound on the log of the marginal data density

$$\begin{aligned} C_q(\mathbf{y}) = E_{q(H)}\Bigg\{ & \sum_g \log p(d_g|b_g,\psi_g,\sigma_g^2,\tau) + \sum_g \log p(m_g|\sigma_g^2) \\ & + \sum_g \log p(\sigma_g^2) - \sum_g \log q(\sigma_g^2) + \log p(\tau) - \log q(\tau) \\ & + \sum_g \log p(\psi_g|\nu,\sigma_g^2) + \log p(\nu) - \sum_g \log q(\psi_g) - \log q(\nu) \\ & + \sum_g \log p(b_g|p) + \log p(p) - \sum_g \log q(b_g) - \log q(p)\Bigg\}. \end{aligned}$$

It is only necessary to update the variational posterior means $\widehat{M_\cdot}$ in the itera-

tive scheme. Upon convergence, the other variational parameters are computed based on the converged value of those involved in the iterations. Therefore, the VB-LIMMA algorithm consists of the following steps:

Step 1: Initialize $\widehat{A_\nu}, \widehat{B_\nu}, \{\widehat{A_{\sigma_g^2}}\}, \{\widehat{B_{\sigma_g^2}}\}, \{\widehat{M_{\psi_g}}\}, \{\widehat{M_{b_g}}\}$.

Step 2: Cycle through $\widehat{M_\tau}, \widehat{B_\nu}, \{\widehat{B_{\sigma_g^2}}\}, \{\widehat{M_{\psi_g}}\}, \{\widehat{M_{b_g}}\}$ iteratively, until the increase in $C_q(\mathbf{y})$ computed at the end of each iteration is negligible.

Step 3: Compute $\widehat{V_\tau}, \{\widehat{V_{\psi_g}}\}, \widehat{\alpha_1}, \widehat{\alpha_0}$ using converged variational parameter values.

CHAPTER 3

# VARIATIONAL BAYES NUMERICAL ILLUSTRATIONS

## 3.1 Simulation

The goal of this simulation study is to assess the performance of VB as a classifier when various model assumptions are correct, and to determine the accuracy of VB approximation of marginal posterior distributions.

### 3.1.1 The B-LIMMA model

Under the B-LIMMA model (2.4), the difference in sample means $\{d_g\}$ and mean squared errors $\{m_g\}$ are simulated according to model (1) of Bar et al. (2010) with $\mu = 0$ and $\gamma_g \equiv 0$. VB-LIMMA approximates a marginal posterior density whose mean is then used as posterior estimate of the corresponding parameter for classification. Performance comparison is made between VB implemented in R and the widely-used `limma` package (Smyth, 2005).

A data set containing $G = 5,000$ genes was simulated under the B-LIMMA model with $n_{1g} \equiv n_{2g} \equiv 8, \tau = 0, \sigma_g^2 \sim IG(41, 400)$ *i.i.d.*, and $\psi_g$ and $b_g$ generated as specified in model (2.4). The shape and scale parameter values in the Inverse-Gamma distribution were chosen such that $\sigma_g^2, g = 1, .., G$, were generated with mean 10 and moderate variation among the $G$ genes. We set $p$, the non-null proportion, and $v$, the variance factor such that $p \in \{0.05, 0.25\}, v \in \{\frac{1}{2}, 2\}$. Simulations with various sets of parameter values were investigated, but these values lead to representative results among our experiments. For classification, a

$N(0, 100)$ prior for $\tau$, $IG(0.1, 0.1)$ prior for $v$ and for $\sigma^2_{\epsilon,g}$, and a $Beta(1, 1)$ prior for the non-null proportion $p$ were used. In the VB-LIMMA algorithm, the error tolerance was set to be $10^{-6}$. The starting values were $\widehat{A_v} = \widehat{B_v} = 0.1, \sigma^2_{\epsilon,g} \equiv 1, \widehat{M_{\psi_g}} \equiv 0$, and $\widehat{M_{b_g}} = 1$ for genes that are associated with the 5% largest values of $d_g$ or the 5% smallest values of $d_g$, and $\widehat{M_{b_g}} = 0$ otherwise. For each gene $g$, $\widehat{M_{b_g}}$ was the VB-approximated posterior mean of the latent indicator $b_g$, i.e. the gene-specific posterior non-null probability. Hence, for detection of non-null genes, gene ranking based on $\widehat{M_{b_g}}$ can be compared with gene ranking based on the B-statistic, or equivalently gene-specific posterior non-null probability computable from the B-statistic, produced by `limma`.

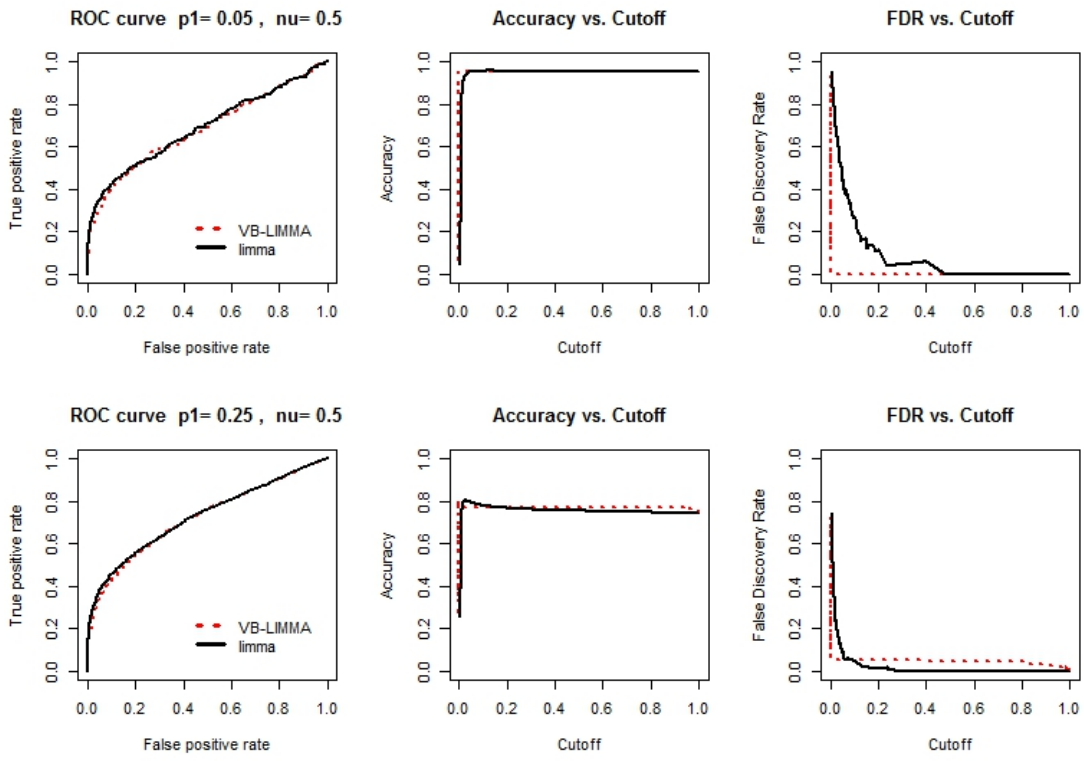


Figure 3.1: `limma` and VB-LIMMA classification comparison on simulated two-sample data with $G = 5,000, n_{1g} \equiv n_{2g} \equiv 8, \tau = 0, p \in \{0.05, 0.25\}, v = \frac{1}{2}, \sigma^2_g \sim IG(41, 400)$ (*i.i.d.*).

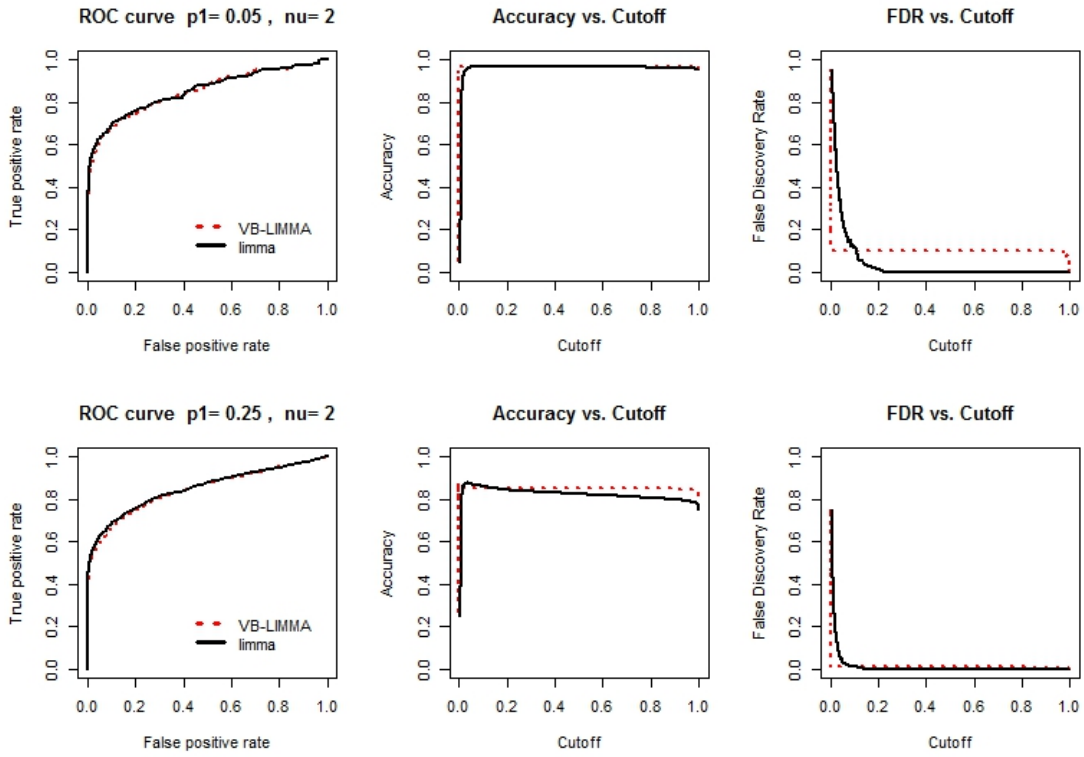


Figure 3.2: `limma` and VB-LIMMA classification comparison on simulated two-sample data with $G = 5,000, n_{1g} \equiv n_{2g} \equiv 8, \tau = 0, p \in \{0.05, 0.25\}, \nu = 2, \sigma_g^2 \sim IG(41, 400)$ (*i.i.d.*).

We note that, varying the value of $\nu$ while keeping $\sigma^2_{\epsilon,g}$ unchanged in simulation varies the level of difficulty of the classification problem based on the simulated data. This is because, the B-LIMMA model specifies that the null component $N(\tau, \sigma^2_{\epsilon,g} c_g)$ differs from the non-null component $N(\tau, \sigma^2_{\epsilon,g} c_g + \nu\sigma^2_{\epsilon,g})$ only in variance, and this difference is governed by the multiplicative factor $\nu$. Therefore, data simulated with large $\nu$ tends to include non-null genes that are far away from mean $\tau$ and thus clearly distinguishable from null genes. The less overlap of the non-null and null component distributions the easier the task of sparse classification for either method. This is reflected in improved ROC curves for both methods as $\nu$ is changed from 0.5 to 2 in Figures 3.1 and

3.2. In fact, in Figures 3.1 and 3.2 we also see that for simulated data with $p \in \{0.05, 0.25\}, \nu \in \{\frac{1}{2}, 2\}$, classification by VB-LIMMA is almost identical to classification by `limma` in terms of ROC curve performance. A cutoff value is defined such that if $\widehat{M_{b_g}} \geq$ cutoff then gene $g$ is classified as non-null. Accuracy is defined as $\frac{TP+TN}{P+N}$ and false discovery rate (FDR) as $\frac{FP}{TP+FP}$, as in Sing et al. (2007). Focusing on a practical range of cutoff values, i.e. $(0.5, 1)$, in Figures 3.1 and 3.2, we see that accuracy and FDR of VB are both close to those of `limma`, and that VB has higher accuracy than `limma`. In summary, our experiment suggests that VB-LIMMA, the fully Bayesian classifier, acts as a comparable classifier to `limma`.

### 3.1.2 A mixture model extended from the LIMMA model

The two-component mixture model in Bar et al. (2010) differs from the LIMMA model in the assumption on the gene-specific treatment effect $\psi_g$, specifically

$$\psi_g|\psi, \sigma_\psi^2 \sim N(\psi, \sigma_\psi^2) \quad (i.i.d.) \tag{3.1}$$

where for the random effect the mean $\psi$ is allowed to be non-zero and the variance $\sigma_\psi^2$ is assumed to be independent of the gene-specific error variance. This model is referred to as the LEMMA model in Bar et al. (2010). The LEMMA and LIMMA models both facilitate simultaneous testing of treatment effects on a large number of genes by "borrowing strength" across the genes. In Bar et al. (2010), an empirical Bayes approach is adopted in which parameter estimation of the prior on the error variance $\sigma_{\epsilon,g}^2$ is accomplished through maximum marginal likelihood and point estimates of the global parameters are obtained via the EM algorithm. In the EM algorithm, evaluation of the complete data likelihood involves integrating over the prior on the error variance $\sigma_{\epsilon,g}^2$ which

is achieved via a Laplace approximation. In what follows we consider a natural extension to the LEMMA model: a fully Bayesian three-component mixture model, B-LEMMA. Formulation of the B-LEMMA model and details of the corresponding VB-LEMMA algorithm are similar to model (2.4) and what is outlined in Section 2.3.2, and are included in Appendix A.1.

The computational method in Bar et al. (2010) for estimation via Laplace approximation and EM algorithm is implemented in the `lemma` R package (Bar and Schifano, 2010). We consider an MCMC procedure for the B-LEMMA model so that the performance of VB-LEMMA for classification is assessed through comparison with MCMC-LEMMA with reference to `lemma`.

A data set containing $G = 5,000$ genes was simulated under the B-LEMMA model using $n_{1g} \equiv 6, n_{2g} \equiv 8, \tau = 0, p_1 = p_2 = 0.1, \psi = 20, \sigma_\psi^2 = 2, \sigma_g^2 \sim IG(41, 400)$ *i.i.d.*, and $\psi_g$ was generated based on (3.1). In the VB-LEMMA algorithm, the error tolerance was set to be $10^{-6}$. Starting values were set under the same scheme as the VB-LIMMA algorithm, except that the parameter $\nu$ was replaced by $\psi$. The posterior mean of $\psi$, $\widehat{M_\psi}$, was initialized as the average of the absolute difference between the mean of $d_g$ and the mean of the 5% largest $d_g$, and the absolute difference between the mean of $d_g$ and the mean of the 5% smallest $d_g$. Non-informative independent priors were assigned to the global parameters, so that $\tau$ and $\psi$ each had a $N(0, 100)$ prior, $\sigma_\psi^2$ and $\sigma_g^2$ each had an $IG(0.1, 0.1)$ prior, and $(p_1, p_2, 1 - p_1 - p_2)$ had a $Dirichlet(1, 1, 1)$ prior. For comparison of classification performance, we used the same prior distributions and starting values in MCMC-LEMMA as in VB-LEMMA. MCMC-LEMMA was run with 1 chain of length 1,200,000, a burn-in period 1,195,000, and a thinning factor 10. We made this choice of chain length to ensure chain convergence after

experimenting with various chain settings.

Because point estimates suffice for classification, we present VB-LEMMA and MCMC-LEMMA classification results using posterior means computed from the approximate posterior marginal densities. Genes are classified with a prescribed cutoff value. That is, based on the mean from the approximate marginal posterior of $(b_{1g}, b_{2g})$ outputted by the VB-LEMMA algorithm, $\widehat{M_{b_{1g}}} \geq$ cutoff $\iff$ gene $g$ belongs to non-null 1 component, and $\widehat{M_{b_{2g}}} \geq$ cutoff $\iff$ gene $g$ belongs to non-null 2 component. The posterior mean of the $(b_{1g}, b_{2g})$ sample for any gene $g$ is calculable from the MCMC samples and is used to classify gene $g$ with the same cutoff. Point estimates outputted by `lemma` run with default starting values are used as a reference for accuracy of point estimates.

On an Intel Core i5-2430M 2.40GHz, 6GB RAM computer, it took `lemma`, VB-LEMMA, and MCMC-LEMMA 20.94 seconds, 29.24 seconds, and 25353.88 seconds (7 hours) to run, respectively. Table 3.1 shows the `lemma` point estimates and the approximate marginal posterior means from VB-LEMMA and from MCMC-LEMMA. The parameter estimates from the three methods are all close to the true values except for $\sigma^2_\psi$. The disagreement between the MCMC-LEMMA and VB-LEMMA estimates of $\sigma^2_\psi$ is also observable in Figure 3.3 where the marginal posterior density plots are displayed. In Figure 3.3, the posterior means of VB-LEMMA and MCMC-LEMMA agree well with each other and with `lemma`, since the vertical lines almost coincide for all parameters except for $\sigma^2_\psi$. Deviation of the MCMC-LEMMA estimate of $\sigma^2_\psi$ 6.993 from the true value 2 renders chain convergence still questionable, despite the long chain we use. Yet we see in Figure 3.4, which contains the mixture and component densities plots, that VB-LEMMA and MCMC-LEMMA appear to identify the correct

non-null 1 and non-null 2 genes since the inward ticks and outward ticks on the x-axis locate at where the two non-null components truly are in the second and fourth plots. True positive rate, TPR, is defined as $\frac{TP1+TP2}{P1+P2}$ and accuracy defined as $\frac{TP1+TP2+TN}{P1+P2+N}$, where $TP1$, $TP2$, $TN$, $P1$, $P2$, and $N$ are, respectively, number of correctly labeled non-null 1 genes, number of correctly labeled non-null 2 genes, number of correctly labeled null genes, number of true non-null 1 genes, number of true non-null 2 genes, and number of true null genes, such that in this example $P1 + P2 + N = G$. Figure 3.4 and Table 3.1 confirm that VB-LEMMA and MCMC-LEMMA classify the genes correctly with high true positive rate and accuracy on the simulated data.

| | true | `lemma` | VB-LEMMA | MCMC-LEMMA |
|---|---|---|---|---|
| $\tau$ | 0 | -0.00650 | 0.00432 | -0.000682 |
| $\psi$ | 20 | 19.764 | 19.850 | 19.737 |
| $\sigma^2_\psi$ | 2 | 5.247 | 4.278 | 6.993 |
| $p_1$ | 0.1 | 0.105 | 0.103 | 0.105 |
| $p_2$ | 0.1 | 0.105 | 0.104 | 0.105 |
| Number of genes in non-null 1 | 525 | 520 | 508 | 510 |
| Number of genes in non-null 2 | 524 | 516 | 518 | 518 |
| TPR | NA | 0.987 | 0.977 | 0.978 |
| Accuracy | NA | 0.997 | 0.995 | 0.995 |

Table 3.1: Posterior estimates, deduced number of non-null genes, and TPR and accuracy for `lemma`, VB-LEMMA, and MCMC-LEMMA on simulated B-LEMMA data with $G = 5,000, n_{1g} \equiv 6, n_{2g} \equiv 8, \sigma^2_g \sim IG(41, 400)$ (*i.i.d.*) and cutoff= 0.8.

The reason why we implement MCMC-LEMMA with only one chain is that we wish to avoid the label-switching issue noted in Section 2.3.1. For MCMC-LEMMA with one chain, should label-switching occur, $\psi$ would have the opposite sign and the non-null proportions $p_1$ and $p_2$ would exchange positions, leading to classification results in which the labels of non-null components 1 and 2 are switched. Thus, with a single chain, the label switching issue is easily resolved. If multiple chains of $\psi$ are run, some will have switched labels. The

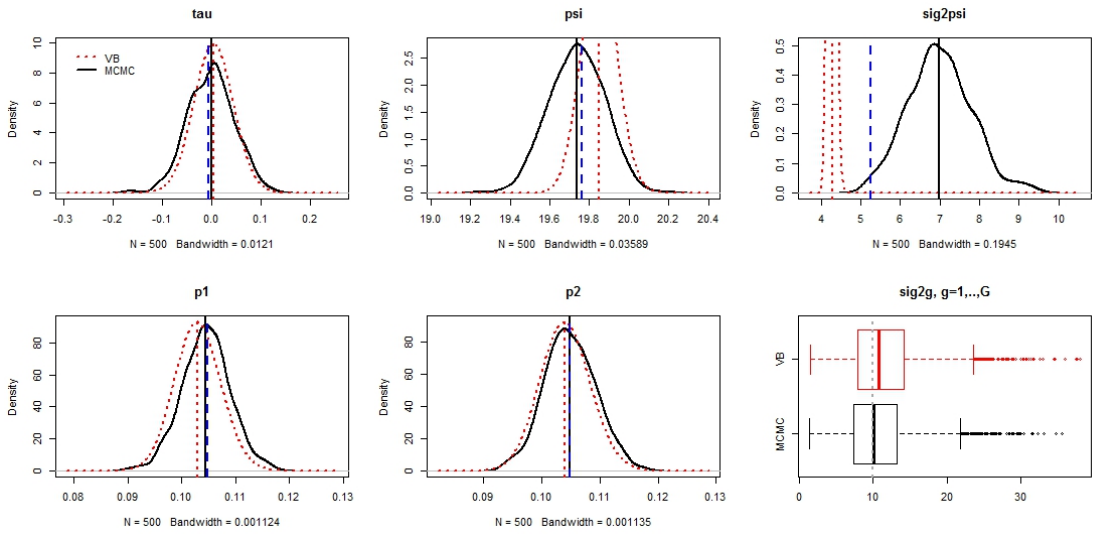


Figure 3.3: Marginal posterior plots approximated by VB-LEMMA and by MCMC-LEMMA, for the parameters in the B-LEMMA model. For each parameter, the dotted and solid vertical lines represent the estimated posterior mean by VB and by MCMC respectively, and the dashed vertical line marks the estimated parameter value given by `lemma`. The bottom right plot shows the posterior mean of the gene-specific error variance $\sigma_g^2$ for each gene $g$, estimated by VB-LEMMA and by MCMC-LEMMA, and how this quantity varies across the G genes. The dotted line indicates the prior mean 10 of $\sigma_g^2$ since the prior distribution of $\sigma_g^2$ are i.i.d. $IG(41, 400)$.

average of the MCMC simulated samples would then be misleading. Although using one chain requires a much longer chain than when multiple chains are used for reaching convergence, it allows us to distinguish label-switching from non-convergence and easily correct for label-switching in MCMC-LEMMA output.

One approach to address label-switching issue in mixture models is post-processing of MCMC solutions. In the B-LEMMA classification problem, once label-switching is detected, we can post-process MCMC-LEMMA classification

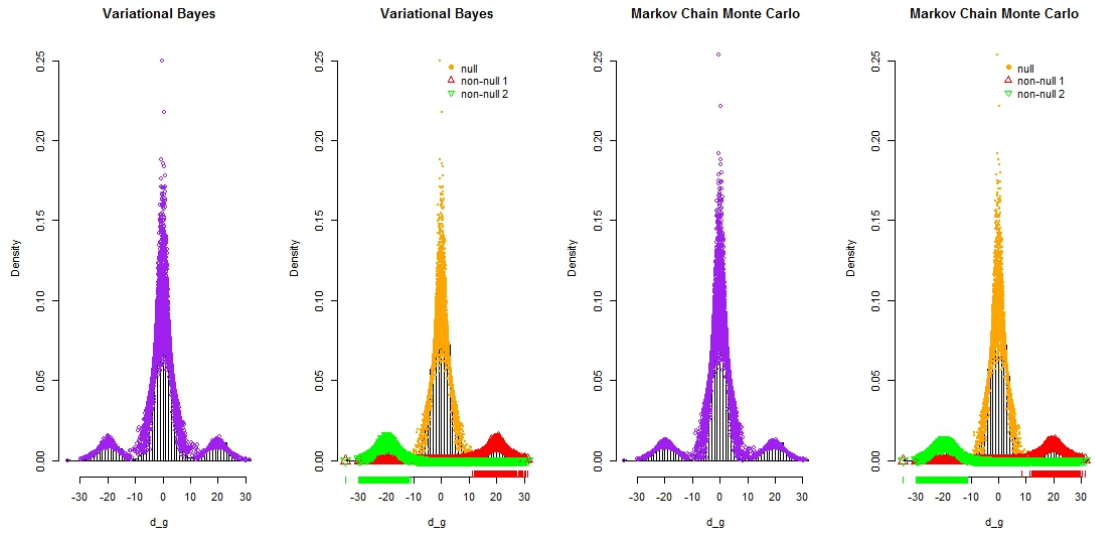


Figure 3.4: Approximated densities on simulated data with $G = 5,000$. From left to right: mixture density approximated by VB-LEMMA, the three component densities approximated by VB-LEMMA, mixture density approximated by MCMC-LEMMA, and the three component densities approximated by MCMC-LEMMA. On the x-axis in the second and the fourth plots, inward ticks indicate the classified non-null 1 genes and outward ticks indicate the classified non-null 2 genes by the corresponding method.

results by giving the opposite sign to the MCMC sample of $\psi$, switching the $p_1$ sample for the $p_2$ sample and vice versa, and exchanging the non-null 1 and non-null 2 statuses. To make Figure 3.5, where performance of VB-LEMMA and MCMC-LEMMA was recorded for 30 simulated data sets, the data simulation settings and the settings for the two methods were kept unchanged from the previous experiment, and MCMC-LEMMA results were post-processed according to the aforementioned scheme wherever label-switching occurred. Posterior means of $\sigma^2_\psi$ estimated by MCMC-LEMMA persisted as worse over-estimates than those estimated by VB-LEMMA. Nonetheless, the component densities plots for the 30 simulated data sets (not shown) with gene labels all appear simi-

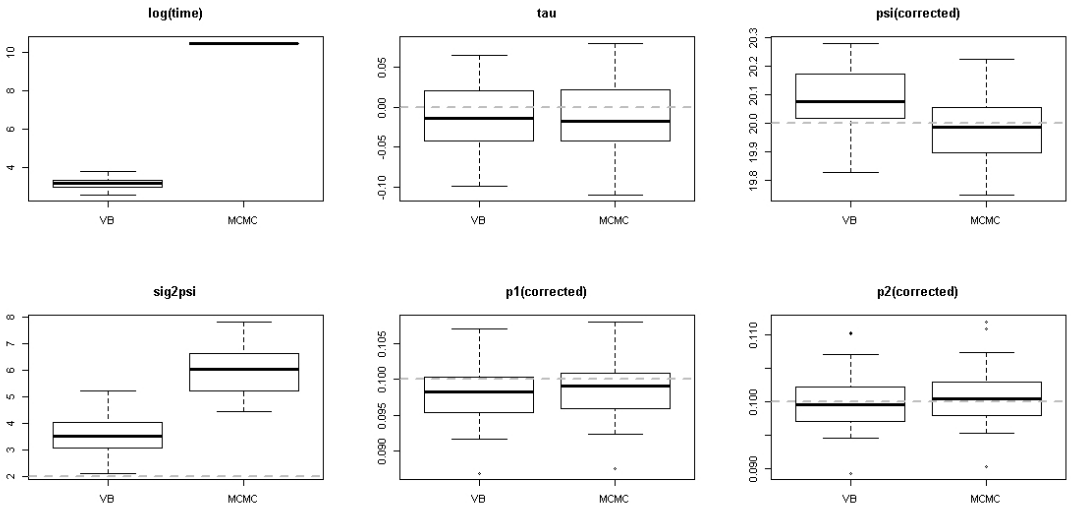


Figure 3.5: Comparison between VB-LEMMA and MCMC-LEMMA with multiple simulation on an Intel Xeon L5410 2.33GHz, 8GB RAM computer. 30 data sets were simulated with same true parameter values in Table 3.1. MCMC-LEMMA results were post-processed so that results were corrected for label-switching among the simulated data sets. The dashed lines mark the true parameter values.

lar to those in Figure 3.4, indicating that both VB-LEMMA and MCMC-LEMMA are reliable classifiers in this context. However, VB-LEMMA is far superior in terms of computational efficiency.

Another way of avoiding label-switching issue in mixture models is assigning a prior distribution on the parameter that appropriately restricts the sampling process in MCMC, as Christensen et al. (2011) points out. Although the mixture and component densities plots (not shown) show that MCMC-LEMMA, when implemented with a half-normal prior on $\psi$, classifies the genes satisfactorily, restricting the mean of $\psi$ to a positive value changes the marginal data distribution of the B-LEMMA model. Therefore, direct comparisons with the modified MCMC-LEMMA are no longer valid because VB-LEMMA and

MCMC-LEMMA with a half-normal prior on $\psi$ are based on two different models.

These B-LEMMA simulation experiments indicate that VB-LEMMA efficiently and accurately approximates fully Bayesian estimation and classification results. In contrast, MCMC-LEMMA, although theoretically capable of achieving extremely high accuracy, requires substantially more work in overcoming issues such as non-convergence and label-switching.

### 3.1.3 A mixture model for count data

In mass-spectrometry-based shotgun proteomics, spectral counts measure abundance of proteins under various conditions. By comparing spectral counts under control and treatment conditions, researchers can test for differential abundance simultaneously for a large number of proteins. Booth et al. (2011) propose a mixture of log-linear models with random protein specific factors for classifying proteins into null and non-null (differential abundance) categories. Their model, which is fully Bayesian, can be written as follows:

$$y_{ij}|\mu_{ij} \sim Poisson(\mu_{ij}), \tag{3.2}$$

$$\log \mu_{ij}|I_i, \beta_0, \beta_1, b_{0i}, b_{1i} = \beta_0 + b_{0i} + \beta_1 T_j + b_{1i} I_i T_j + \beta_2 I_i T_j + \log L_i + \log N_j,$$

$$I_i|\pi_1 \sim Bernoulli(\pi_1) \;\; i.i.d., \qquad b_{ki}|\sigma_k^2 \sim N(0, \sigma_k^2) \;\; i.i.d., \;\; k = 0, 1,$$

$$\beta_m \sim N(0, \sigma_{\beta_m}^2), \;\; m = 0, 1, 2, \qquad \sigma_k^{-2} \sim Gamma(A_{\sigma_k^2}, B_{\sigma_k^2}), \;\; k = 0, 1, \qquad \pi_1 \sim Beta(\alpha, \beta),$$

where $y_{ij}$ is the spectral count of protein $i$, $i = 1, .., p$, and replicate $j$, $j = 1, .., n$. $L_i$ is the length of protein $i$, $N_j$ is the average count for replicate $j$ over all proteins, and

$$T_j = \begin{cases} 1 & \text{if replicate } j \text{ is in the treatment group} \\ 0 & \text{if replicate } j \text{ is in the control group.} \end{cases}$$

In fact, conjugacy in this Poisson GLMM is not sufficient for a tractable solution to be computed by VB. Therefore, a similar Poisson-Gamma HGLM where the parameters $\beta_m, m = 0, 1, 2$ and the latent variables $b_{ki}, k = 0, 1$ are transformed is used for the VB implementation:

$$\begin{aligned} y_{ij}|\mu_{ij} &\sim Poisson(\mu_{ij}), \qquad (3.3) \\ \log \mu_{ij}|I_i, \beta_0, \beta_1, b_{0i}, b_{1i} &= \beta_0 + b_{0i} + \beta_1 T_j + b_{1i} I_i T_j + \beta_2 I_i T_j + \log L_i + \log N_j, \\ I_i|\pi_1 &\sim Bernoulli(\pi_1) \;\; i.i.d., \qquad b_{ki}|\phi_{ki} = \log(\phi_{ki}^{-1}), \;\; k = 0, 1, \\ \phi_{ki}|\delta_k &\sim IG(\delta_k, \delta_k) \;\; i.i.d., \; k = 0, 1, \qquad \delta_k \sim Gamma(A_{\delta_k}, B_{\delta_k}), \;\; k = 0, 1, \\ \beta_m|\lambda_m = \log(\lambda_m^{-1}), \;\; m &= 0, 1, 2, \quad \lambda_m \sim IG(A_{\lambda_m}, B_{\lambda_m}), \;\; m = 0, 1, 2, \quad \pi_1 \sim Beta(\alpha, \beta). \end{aligned}$$

As before classification is inferred from the posterior expectations of the latent binary indicators $I_i, i = 1, ..., p$. This fully Bayesian, two-component mixture model allows for derivation of a VB algorithm, VB-proteomics, the details of which are shown in supplementary materials.

For any real data, the exact model that the data distribution follows is unknown. To imitate challenges in sparse classification on real data sets, we simulated data from each of the two models: Poisson GLMM (3.2) and Poisson-Gamma HGLM (3.3), and applied VB-proteomics to both data sets. MCMC-proteomics, also derived based on Poisson-Gamma HGLM (3.3), was implemented via OpenBUGS in R with a chain of length 600,000, a burn-in period 595,000, and a thinning factor 5. Starting values implemented in VB-proteomics were consistent with those in MCMC-proteomics.

Two data sets, one following the Poisson GLMM (3.2) with $\beta_0 = -7.7009$, $\beta_1 = -0.1765, \beta_2 = 0, \sigma_0^2 = 1, \sigma_1^2 = 4$ and the other following the Poisson-Gamma HGLM (3.3) with $\lambda_0 = 2210.336$, $\lambda_1 = 1.193$, $\lambda_2 = 1$, $\delta_0 = 1$, $\delta_1 = 0.2$ were generated. In each data set there were 4 replicates under each treatment and $p = 1307$ proteins. The non-null indicator was generated with $\pi_1 = 0.2$. These values were chosen so that the simulated data sets were similar to the Synthetic 2-fold data analyzed in Booth et al. (2011).

| | Poisson GLMM | | | Poisson-Gamma HGLM | | |
|---|---|---|---|---|---|---|
| | true | VB | MCMC | true | VB | MCMC |
| number of non-nulls | 259 | 109 | 96 | 262 | 103 | 90 |
| $\pi_1$ | 0.2 | 0.0856 | 0.160 | 0.2 | 0.122 | 0.218 |
| accuracy | | 0.878 | 0.874 | | 0.871 | 0.865 |
| time | | 54 seconds | 7.5 hours | | 10 seconds | 7.4 hours |

Table 3.2: Proteomics count data: simulation under the two models and VB-proteomics and MCMC-proteomics classifications. Total number of proteins is 1,307. Cutoff is 0.8. Accuracy is as defined in Section 3.1.1.

Table 3.2 gives the classification and computational performance of VB-proteomics and MCMC-proteomics for the two simulated data sets on an Intel Core i5-2430M 2.40GHz, 6GB RAM computer. The longer running time and the larger deviation of posterior mean of $\pi_1$ from its true value associated with the simulated data under Poisson GLMM in Table 3.2 shows that it is indeed more difficult for VB-proteomics to perform classification on data from the "wrong" model. Nonetheless, VB-proteomics identifies the extreme proteins for both data sets with similar accuracy regardless of which model the data was generated from, and is much more desirable than MCMC-proteomics in terms of computational speed.

Figures 3.6 and 3.7 show plots of the log ratios of total counts (+1) in treatment and control groups against protein number, and ROC curves compar-

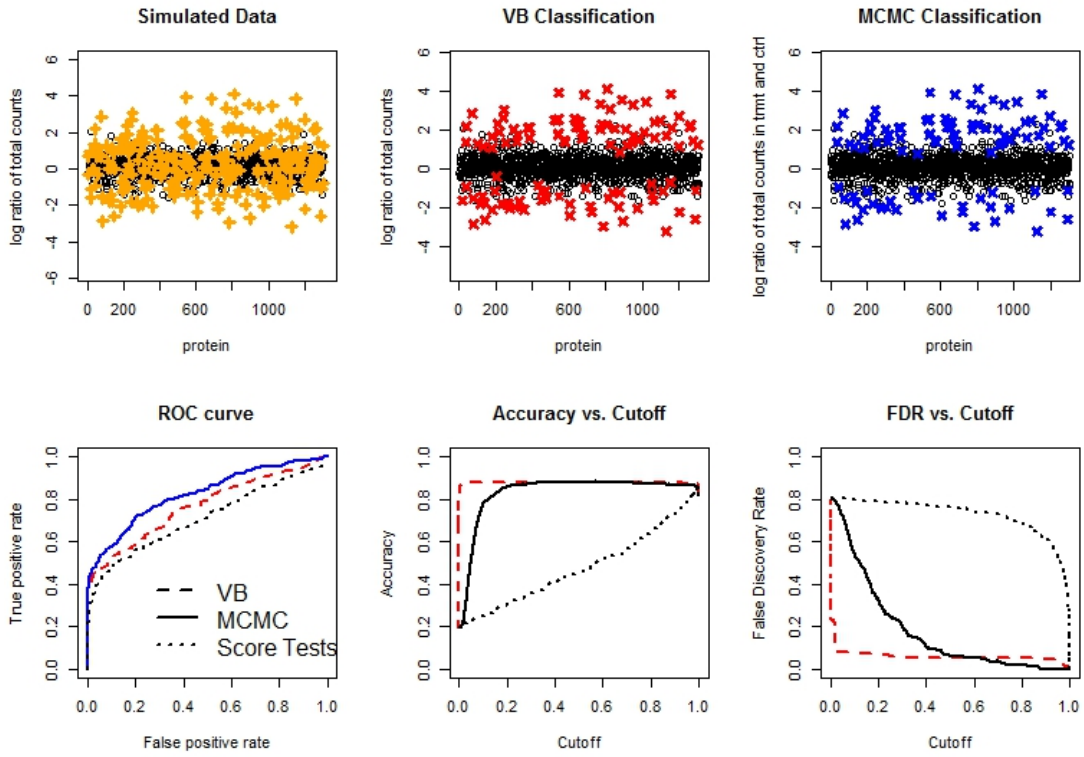


Figure 3.6: Simulated data under the Poisson GLMM (3.2), i.e. the "wrong" model. Top row: plots of log ratios of total counts in treatment and control groups against protein number with true non-null proteins, VB-proteomics labeled proteins, and MCMC-proteomics labeled proteins indicated. Bottom row: ROC curves, accuracy vs. cutoff, and FDR vs. cutoff for VB-proteomics, MCMC-proteomics, and for one protein at a time score tests.

ing performance of the model-based VB-proteomics, MCMC-proteomics approaches with one protein at a time Score tests. These plots clearly indicate that VB-proteomics acts as a better classifier than individual Score tests and as a comparable classifier to MCMC-proteomics in terms of ROC curve, accuracy, and false discovery rate performance.

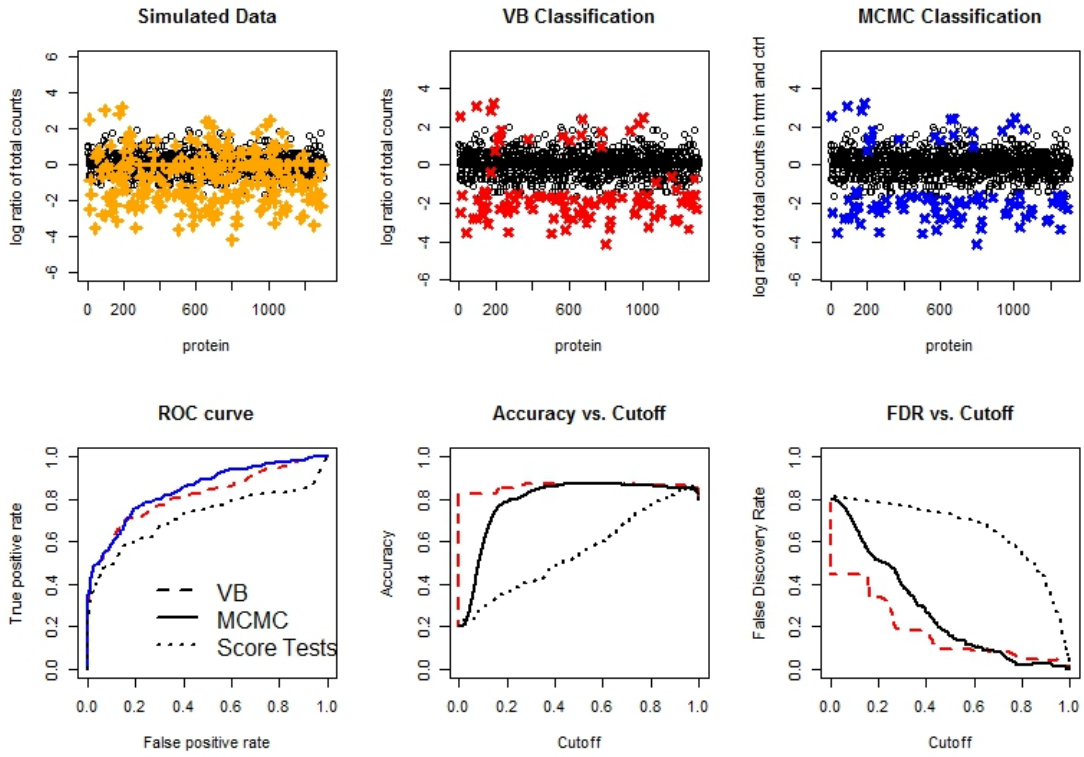


Figure 3.7: Simulated data under the Poisson-Gamma HGLM (3.3). Top row: plots of log ratios of total counts in treatment and control groups against protein number with true non-null proteins, VB-proteomics labeled proteins, and MCMC-proteomics labeled proteins indicated. Bottom row: ROC curves, accuracy vs. cutoff, and FDR vs. cutoff for VB-proteomics, MCMC-proteomics, and for one protein at a time score tests.

## 3.2 Real data examples

In this section, we further illustrate classification performance of VB for fully Bayesian finite mixture models on real data. Two microarray examples are investigated, the APOA1 (Callow et al., 2000) and Colon Cancer (Alon et al., 1999) data sets.

### 3.2.1 APOA1 data

The APOA1 data (Callow et al., 2000) contains 5,548 genes associated with measurements from 8 control mice and 8 "knockout" mice. The data was originally obtained from a two-groups design. The histogram of $d_g$ in Figure 3.8 indicates that the B-LIMMA model (2.4) is a good fit and that both `limma` and VB-LIMMA are applicable for classification, under the assumption that majority of the genes are null. The small proportion of the non-null genes is known since there are eight true non-null genes. Figure 3.9 shows the top 8 non-null genes by `limma`. VB-LIMMA was implemented, with non-informative priors and starting values such that $\widehat{M_{b_g}} = 1$ for genes that are associated with the 5% largest values of $d_g$ or the 5% smallest values of $d_g$ and $\widehat{M_{b_g}} = 0$ otherwise. VB-LIMMA identified the same top eight genes as `limma` in 53 seconds, indicated by close-to-one posterior non-null probability estimate of those genes in Figure 3.8. The `convest` function in `limma` gave an over-estimate of the non-null proportion $p$, 0.130114, whereas VB-LIMMA produced a closer estimate 0.00162162 to the true value 0.00144.

### 3.2.2 Colon Cancer data

The Colon Cancer data (Alon et al., 1999) consists of gene expression values from 2,000 genes in 22 controls and 40 treatment samples. Since the two-component mixture model is a special case of the three-component mixture model with one of the non-null proportions being zero, both B-LIMMA and B-LEMMA models are suitable for classification of genes on the Colon Cancer data. Therefore, `limma`, VB-LIMMA, `lemma`, and VB-LEMMA were imple-

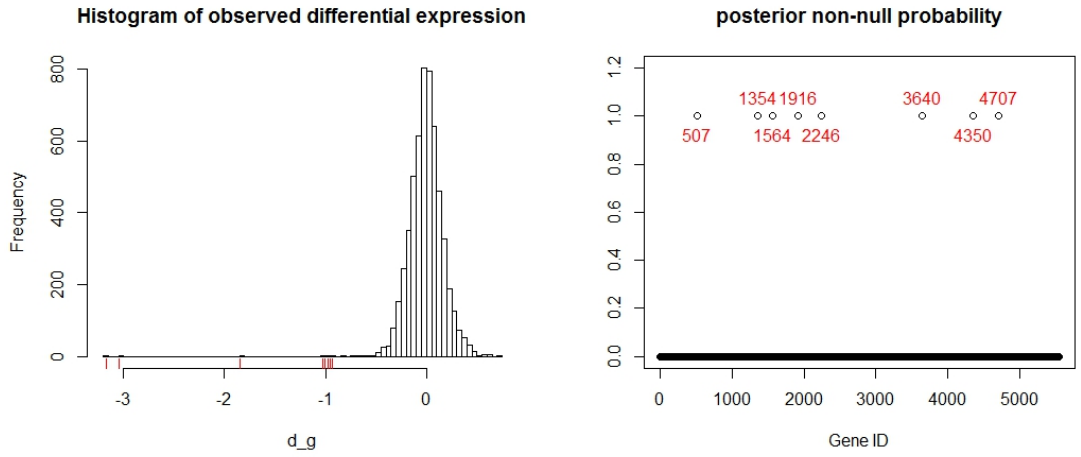


Figure 3.8: APOA1 data: histogram of $d_g$ with the eight non-null genes classified by VB-LIMMA indicated by inward ticks on the x-axis, and posterior mean of $b_g$ (i.e. posterior non-null probability of gene $g$) estimated by VB-LIMMA with the eight non-null gene numbers identified.

| | NAME | logFC | t | P.Value | adj.P.Val | B |
|---|---|---|---|---|---|---|
| 1916 | ApoAI,lipid-Img | -3.1661645 | -24.048796 | 5.442072e-15 | 3.019261e-11 | 15.960594 |
| 507 | EST,HighlysimilartoA | -3.0485504 | -12.938819 | 1.797129e-10 | 4.985235e-07 | 11.347701 |
| 4707 | CATECHOLO-METHYLTRAN | -1.8481659 | -12.458569 | 3.300689e-10 | 6.104074e-07 | 10.982250 |
| 3640 | EST,WeaklysimilartoC | -1.0269537 | -11.894827 | 6.905520e-10 | 9.577957e-07 | 10.524492 |
| 1564 | ApoCIII,lipid-Img | -0.9325824 | -9.927515 | 1.145390e-08 | 1.270925e-05 | 8.647893 |
| 2246 | ESTs,Highlysimilarto | -1.0098117 | -9.065277 | 4.461600e-08 | 3.579128e-05 | 7.667271 |
| 1354 | est | -0.9774236 | -9.057852 | 4.515842e-08 | 3.579128e-05 | 7.658359 |
| 4350 | similartoyeaststerol | -0.9549693 | -7.465666 | 7.068872e-07 | 4.902263e-04 | 5.546823 |

Figure 3.9: `limma` output: Top 8 genes detected as non-null for APOA1 data. Gene IDs are shown in the first column.

mented.

A gene rankings table showing genes that are associated with the top 200 largest posterior non-null probabilities was produced according to results from each of the four procedures. For all procedures, the posterior non-null probability for any gene equals one minus the posterior null probability of that gene. Comparing the top genes tables reveals that out of the top 200 identified non-

null genes, `limma` agrees with VB-LIMMA on 151 genes, `lemma` agrees with VB-LEMMA on 190 genes, and all four methods share 140 genes. This high level of agreement between the procedures shows that fully Bayesian classification via VB is comparable in accuracy to classification via empirical Bayes approaches on the Colon Cancer data.

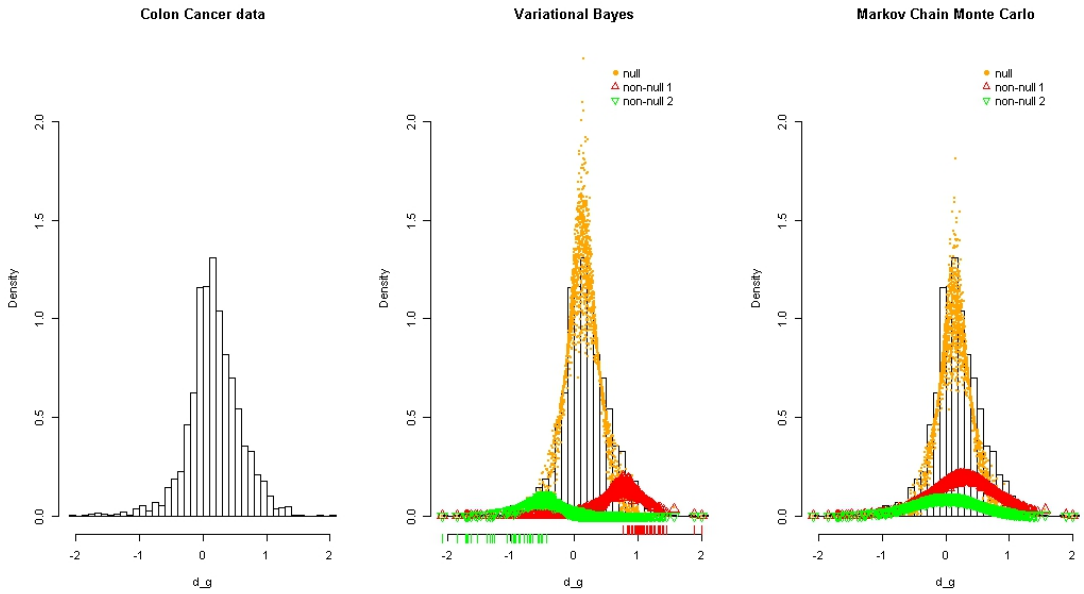


Figure 3.10: From left to right: Histogram of $d_g$ for Colon Cancer data, component densities estimated by VB-LEMMA, and component densities estimated by MCMC-LEMMA. On the x-axis, inward ticks indicate the classified non-null 1 genes and outward ticks indicate the classified non-null 2 genes with a 0.9 cutoff.

MCMC-LEMMA was also implemented for the Colon Cancer data to illustrate performance of MCMC for classification on real data under the B-LEMMA model. A single chain of length 4,000,000, a burn-in period 3,995,000, and a thinning factor 10 was used to ensure convergence and avoid label-switching. MCMC diagnostics plots (not shown) suggests that convergence is tolerable, although it took MCMC-LEMMA 11.7 hours to run on an Intel Xeon L5410

2.33GHz, 8GB RAM computer. There are 187 genes present in both top 200 genes tables based on MCMC-LEMMA and VB-LEMMA classifications, which shows accuracy of VB in classification is comparable with that of MCMC. From the component densities plots in Figure 3.10 we see that, the three-component mixture evaluated by VB-LEMMA and by MCMC-LEMMA are two different yet plausible solutions based on the B-LEMMA model. It is unknown which solution is closer to the truth. Nonetheless, with a 0.9 cutoff, MCMC-LEMMA did not detect any non-null genes, whereas VB-LEMMA identified 56 group 1 non-null genes and 30 group 2 non-null genes in 7 seconds.

CHAPTER 4

# MODEL-BASED CLASSIFICATION WITH SIMULTANEOUS ADJUSTMENT FOR UNWANTED VARIATION

## 4.1 Extension of the B-LIMMA model

Many problems in bioinformatics concern association detection which can be regarded as a special case of classification. For example, detecting significant gene expression and genotype association in eQTL mapping is essentially classifying which genes are non-null with respect to any given genotype and which ones are null. Inferential approach as in Smyth (2004) and Bar et al. (2010) can be extended to solve association detection problems for high-dimensional data that contain unwanted variation, with an addition of a factor analysis term that consists of the unknown hidden factors and loadings to the hierarchical mixture model. In the new approach, the hierarchical mixture model is an extension of the B-LIMMA model (2.4) but with gene-level measurement on all groups as the response instead of gene-level difference in group means. Empirical Bayes inference is implemented via an EM algorithm with classification based on the posterior expectation of the latent indicator variables. The hierarchical mixture model specification is similar to HEFT in Gao et al. (2014), but for a mixture structure and some assumptions on prior distributions.

In this dissertation, we discuss model-based classification with simultaneous adjustment for unwanted variation while referring to the response as "gene expression" and the biological variable of interest as "genotype" as in the context of eQTL analysis. However, our approach is applicable to important applications in bioinformatics besides eQTL analysis, such as differential expression

analysis on microarray data or on RNA-sequencing data and xQTL analysis, where the response is not necessarily gene expression data and a grouping variable (e.g. a dummy variable indicating treatment status) is the primary biological variable. Also, when we refer to unwanted variation in the model-based perspective, we use "unwanted variation" and "hidden effects" interchangeably in what follows.

## 4.2 Model-Based Classification with Simultaneous Adjustment for Unwanted Variation

### 4.2.1 RRmix Model

#### Model specification

The HEFT model in Gao et al. (2014) for eQTL analysis is

$$\begin{aligned}
\mathbf{Y} &= \mu \mathbf{1}_G^T + X\boldsymbol{\beta} + X_c\boldsymbol{\gamma} + \Lambda\mathbf{F} + \mathbf{W} \\
\mathbf{W} &= [W_1,\ W_2,\ \ldots,\ W_G], \quad W \sim \mathcal{N}(0, \sigma^2 I_n \otimes I_G) \\
\mathbf{F} &= [F_1,\ F_2,\ \ldots,\ F_G], \quad F \sim \mathcal{N}(0, I_q \otimes I_G) \\
\boldsymbol{\beta} &= [\beta_1,\ \beta_2,\ \ldots,\ \beta_G], \quad \beta \sim \mathcal{N}(0, I_2 \otimes I_G) \\
\boldsymbol{\gamma} &= [\gamma_1,\ \gamma_2,\ \ldots,\ \gamma_G], \quad \gamma \sim \mathcal{N}(0, I_r \otimes I_G),
\end{aligned}$$

where $\mathbf{Y}$ is the $n$ by $G$ matrix of $G$ gene expression variables (standardized) on $n$ samples, $\mu$ is a vector of sample-level mean over genes, $X = [\mathbf{1}_n, \mathbf{x}]$ is an $n$ by 2 known matrix in which $\mathbf{x}$ is an indicator for minor allele presence in a single SNP (standardized), $X_c$ is the $n$ by $r$ matrix of $r$ known covariates (standardized), and

$\mathbf{F}$ ($q \times G$) and $\Lambda$ ($n \times q$) are factors and loadings of the factor analysis component that represents hidden effects.

We built a hierarchical mixture model named RRmix using similar notation as in HEFT, but with $\mathbf{Y}$, $\mathbf{x}$, and $X_c$ each referring to the original gene expression matrix, SNP vector, and matrix of covariates. Furthermore, the RRmix model specification differs to the HEFT model specification in that the RRmix model contains a mixture structure, its error variance is specified to be gene-specific, and the covariate effects are modeled as fixed effects. Detailed connections between HEFT and RRmix are reviewed in Section 4.3. To illustrate the mixture structure, the per gene RRmix model is given by

$$\begin{aligned}
y_g \big| \beta_g, F_g, \sigma_g^2 &= \mu + X\beta_g + X_c\gamma_g + \Lambda F_g + W_g \\
W_g \big| \sigma_g^2 &\sim N_n(0, \sigma_g^2 I_n) \\
F_g &\sim N_q(0, I_q) \\
\beta_g \big| b_g &\sim N_2\left( \begin{bmatrix} 0 \\ b_g\psi \end{bmatrix}, (1-b_g)\begin{bmatrix} \sigma_0^2 & 0 \\ 0 & 0 \end{bmatrix} + b_g \begin{bmatrix} \sigma_0^2 & 0 \\ 0 & \sigma_1^2 \end{bmatrix} \right) \\
b_g &\sim Bernoulli(p) \\
\sigma_g^2 &\sim IG(A, B),
\end{aligned} \tag{4.1}$$

where $y_g$ and $\beta_g = \begin{bmatrix} \beta_{0g} \\ \beta_{1g} \end{bmatrix}$ are the $g$-th column in $\mathbf{Y}$ and $\boldsymbol{\beta}$, respectively, and $IG(A, B)$ denotes an Inverse-Gamma distribution with shape $A$ and scale $B$.

The gene-specific latent indicator $b_g$ allows the conditional distribution of $\beta_g$ given $b_g$ to imply that

$$\begin{aligned}
\beta_{1g} \big| b_g = 0 &\equiv 0 \\
\beta_{1g} \big| b_g = 1 &\sim N\left(\psi, \sigma_1^2\right)
\end{aligned}$$

for each gene $g$. The goal is to estimate the posterior mean of $b_g$ for each gene $g$ with a given SNP. In the context of eQTL analysis, we repeat the process but

with a different SNP each time until we have $G \times N$ such estimates, $N$ being the number of SNPs. That is, implementation is done one-SNP at a time for all genes. Classification of genes given a SNP is made via the estimated posterior mean of $b_g$ for each gene $g$. A threshold $p_1$, e.g. $p_1 = 0.9$, on $\hat{b}_g$ is chosen so that any gene $g$ with $\hat{b}_g \geq p_1$ is declared associated with the given SNP $\mathbf{x}$.

**Shrinkage estimation of error variance**

We adopt the method in Bar et al. (2010) for producing shrinkage estimate of the gene-specific error variance $\{\sigma_g^2\}$. We first obtain gene-specific mean squared error $\{m_g\}$ from the linear model of $y_g$ on fixed covariates that include the given SNP $\mathbf{x}$ and the other covariates as specified by $X_c$. That is, no hidden effect is specified in this simple model. With $X_a = [X, X_c]$ where $X$ and $X_c$ are as defined above and $\eta_g = \begin{bmatrix} \beta_g \\ \gamma_g \end{bmatrix}$, the simple linear regression model is

$$y_g = X_a \eta_g + W_g \tag{4.2}$$

$$W_g \sim N_n(\mathbf{0}, \sigma_g^2 I_n).$$

Residuals form fitting the model for a given gene $g$ give rise to the gene-specific pooled variance estimate $m_g$ for that gene. $m_g$ calculated from observed data approximately follows a chi-squared distribution

$$m_g | \sigma_g^2 \sim \sigma_g^2 \frac{\chi^2_{f_g}}{f_g} \tag{4.3}$$

where $f_g = n - rank(X_a)$ for all $g$. We describe the distribution as approximate because the model does not account for possible unwanted variation in the data. That said, (4.3) for each gene suffices to give rise to a shrinkage estimate of $\{\sigma_g^2\}$ via maximum likelihood. That is, maximum likelihood estimates of hyperparameters $A, B$ of prior distribution $\sigma_g^2 \sim IG(A, B)$ are obtained by maximizing the

marginal likelihood of $\{m_g\}$ via the `nlminb` function in R as in Bar et al. (2010). Then, the posterior mode of $\sigma_g^2$ given $m_g$

$$\hat{\sigma}_g^2 = \left(\frac{f_g/2}{f_g/2+\hat{A}+1}\right)m_g + \left(\frac{\hat{A}+1}{f_g/2+\hat{A}+1}\right)\left(\frac{\hat{B}}{\hat{A}+1}\right) \tag{4.4}$$

can be used as shrinkage estimate of $\sigma_g^2$.

**EM algorithm for RRmix**

Implementation of RRmix is accomplished via the EM algorithm. Key steps in the algorithm are outlined below. See the appendix for detailed derivations. Denote $A_0 = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}$ and $\Psi_\beta = \begin{bmatrix} \sigma_0^2 & 0 \\ 0 & \sigma_1^2 \end{bmatrix}$, so that $A_0\Psi_\beta = \begin{bmatrix} \sigma_0^2 & 0 \\ 0 & 0 \end{bmatrix}$.

i). Initialize $\mu$, $\sigma_0^2$ , $\sigma_1^2$ , $\psi$, $\{\gamma_g\}$, $\Lambda$, $p$.

ii). Obtain $\{m_g\}$ from linear model (4.2) with response $\{y_g\}$, so that$\{\sigma_g^2\}$ is initialized to its shrinkage estimate (4.4).

iii). E-step: Conditional on the current estimate of $\{\sigma_g^2\}$ and the current parameter values, update gene-specific quantities related to $\{b_g\}$, $\{\beta_g\}$, and $\{F_g\}$:

$$\hat{b}_g = \mathbb{E}(b_g|y_g) = \frac{u_g}{u_g+v_g}, \tag{4.5}$$

where $u_g = p \times N_n(y_g;\ \mu + X_c\gamma_g + \mathbf{x}\psi,\ X\Psi_\beta X^T + \Lambda\Lambda^T + \sigma_g^2 I_n)$ and $v_g = (1-p)\times N_n(y_g;\ \mu + X_c\gamma_g,\ XA_0\Psi_\beta X^T + \Lambda\Lambda^T + \sigma_g^2 I_n)$, with $N_n(y_g;\mu_g,\tau_g)$ denoting the multivariate normal density evaluated at $y_g$ with mean $\mu_g$ and covariance $\tau_g$. Also,

$$\begin{aligned}
\hat{\beta}_g^{(1)} &= \Psi_\beta X^T[X\Psi_\beta X^T + \Lambda\Lambda^T + \sigma_g^2 I_n]^{-1}(y_g - \mu - X_c\gamma_g - \mathbf{x}\psi) \\
\hat{\beta}_g^{(0)} &= A_0\Psi_\beta X^T[XA_0\Psi_\beta X^T + \Lambda\Lambda^T + \sigma_g^2 I_n]^{-1}(y_g - \mu - X_c\gamma_g) \\
\hat{F}_g^{(1)} &= \Lambda^T[X\Psi_\beta X^T + \Lambda\Lambda^T + \sigma_g^2 I_n]^{-1}(y_g - \mu - X_c\gamma_g - \mathbf{x}\psi) \\
\hat{F}_g^{(0)} &= \Lambda^T[XA_0\Psi_\beta X^T + \Lambda\Lambda^T + \sigma_g^2 I_n]^{-1}(y_g - \mu - X_c\gamma_g)
\end{aligned} \tag{4.6}$$

$$\hat{\Sigma}^{(1)}_{g11} = \Psi_\beta - \Psi_\beta X^T [X\Psi_\beta X^T + \Lambda\Lambda^T + \sigma_g^2 I_n]^{-1} X\Psi_\beta$$

$$\hat{\Sigma}^{(0)}_{g11} = A_0\Psi_\beta - A_0\Psi_\beta X^T [XA_0\Psi_\beta X^T + \Lambda\Lambda^T + \sigma_g^2 I_n]^{-1} XA_0\Psi_\beta$$

$$\hat{\Sigma}^{(1)}_{g21} = -\Lambda^T [X\Psi_\beta X^T + \Lambda\Lambda^T + \sigma_g^2 I_n]^{-1} X\Psi_\beta$$

$$\hat{\Sigma}^{(0)}_{g21} = -\Lambda^T [XA_0\Psi_\beta X^T + \Lambda\Lambda^T + \sigma_g^2 I_n]^{-1} XA_0\Psi_\beta$$

$$\hat{\Sigma}^{(1)}_{g22} = I_q - \Lambda^T [X\Psi_\beta X^T + \Lambda\Lambda^T + \sigma_g^2 I_n]^{-1} \Lambda$$

$$\hat{\Sigma}^{(0)}_{g22} = I_q - \Lambda^T [XA_0\Psi_\beta X^T + \Lambda\Lambda^T + \sigma_g^2 I_n]^{-1} \Lambda\,.$$

iv). M-step: Conditional on the current estimate of $\{\sigma_g^2\}$ and the current E-step estimates, update $\mu$, $\sigma_0^2$ , $\sigma_1^2$ , $\psi$, $\{\gamma_g\}$, $\Lambda$, $p$:

$$\hat{\mu} = \frac{1}{\sum_g \left(\frac{1}{\sigma_g^2}\right)} \sum_g \frac{1}{\sigma_g^2} \left(y_g - X_c\gamma_g - \hat{b}_g \left(\mathbf{x}\psi + X\hat{\beta}^{(1)}_g + \Lambda\hat{F}^{(1)}_g\right) - \left(1 - \hat{b}_g\right)\left(X\hat{\beta}^{(0)}_g + \Lambda\hat{F}^{(0)}_g\right)\right)$$

$$\hat{\sigma}_0^2 = \frac{1}{G} \sum_g \left(\hat{b}_g \left(\left[\hat{\beta}^{(1)}_g\right]_1^2 + \left[\hat{\Sigma}^{(1)}_{g11}\right]_{11}\right) + \left(1 - \hat{b}_g\right)\left(\left[\hat{\beta}^{(0)}_g\right]_1^2 + \left[\hat{\Sigma}^{(0)}_{g11}\right]_{11}\right)\right)$$

$$\hat{\sigma}_1^2 = \frac{1}{\left(\sum_g \hat{b}_g\right)} \sum_g \left(\hat{b}_g \left(\left[\hat{\beta}^{(1)}_g\right]_2^2 + \left[\hat{\Sigma}^{(1)}_{g11}\right]_{22}\right)\right)$$

$$\hat{\psi} = \frac{1}{\sum_g \left(\frac{\hat{b}_g}{\sigma_g^2}\mathbf{x}^T\mathbf{x}\right)} \sum_g \left\{\frac{\hat{b}_g}{\sigma_g^2}\left(y_g - \mu - \left(X\hat{\beta}^{(1)}_g + \Lambda\hat{F}^{(1)}_g\right)\right)^T \mathbf{x}\right\}$$

$$\hat{\gamma}_g = \left(X_c^T X_c\right)^{-1} X_c^T \left(y_g - \mu - \hat{b}_g \left(\mathbf{x}\psi + X\hat{\beta}^{(1)}_g + \Lambda\hat{F}^{(1)}_g\right) - \left(1 - \hat{b}_g\right)\left(X\hat{\beta}^{(0)}_g + \Lambda\hat{F}^{(0)}_g\right)\right)$$

$$\hat{\Lambda} = \left[\sum_g \frac{1}{\sigma_g^2}\left(\hat{b}_g \left(\left(y_g - \mu - X_c\gamma_g - \mathbf{x}\psi - X\hat{\beta}^{(1)}_g\right)\hat{F}^{(1)T}_g - \frac{1}{2}X\left(\hat{\Sigma}^{(1)\ T}_{g21} + \hat{\Sigma}^{(1)}_{g12}\right)\right)\right.\right.$$
$$\left.\left. + \left(1 - \hat{b}_g\right)\left(\left(y_g - \mu - X_c\gamma_g - X\hat{\beta}^{(0)}_g\right)\hat{F}^{(0)T}_g - \frac{1}{2}X\left(\hat{\Sigma}^{(0)\ T}_{g21} + \hat{\Sigma}^{(0)}_{g12}\right)\right)\right)\right]$$
$$\left[\sum_g \frac{1}{\sigma_g^2}\left(\hat{b}_g\left(\hat{F}^{(1)}_g\hat{F}^{(1)T}_g + \hat{\Sigma}^{(1)}_{g22}\right) + \left(1 - \hat{b}_g\right)\left(\hat{F}^{(0)}_g\hat{F}^{(0)T}_g + \hat{\Sigma}^{(0)}_{g22}\right)\right)\right]^{-1}$$

$$\hat{p} = \frac{1}{G}\sum_g \hat{b}_g\,.$$

v). Repeat the cycle of ii), iii), and iv) until convergence, and in each cycle at ii) use the "adjusted" response $y_g^{adj} = y_g - \hat{\Lambda}\hat{F}_g$ to update $\{\sigma_g^2\}$.

An initial value of $\Lambda$ is obtained via spectral decomposition. Let $X_a = [X, X_c]$, $H = X_a(X_a^T X_a)^{-1} X_a^T$ and $S_g = (I_n - H) y_g y_g^T (I_n - H)$. Then $S = \frac{1}{G} \sum_g S_g$ is an estimate of the covariance matrix of $Y$, i.e.

$$S \approx \Lambda\Lambda^T + \hat{\sigma}^2 I_n$$

where $\hat{\sigma}^2 = \frac{1}{G} \sum_g \hat{\sigma}_g^2$. Let $\Delta_q$ be the $n \times q$ matrix of eigenvectors of $S - \hat{\sigma}^2 I_n$ that correspond to the $q$ largest eigenvalues $a_1, ..., a_q$, and

$$A_q = \text{diag}\left( \sqrt{a_k},\ k = 1, \ldots, q \right). \tag{4.7}$$

Then we set $\Lambda = \Delta_q A_q$ as initial value of the hidden factor loading matrix $\Lambda$.

### 4.2.2 The family of models

We consider the RRmix model as well as models which are simplifications of RRmix without any mixture structure that adjust for unwanted variation but incur less computational burden than RRmix. We denote these models by FF, FR, RF, and RR, where the names indicate whether the primary effect and the error variance are modeled as random or fixed in the context of a linear mixed model. This naming convention is analogous to the model names in Table 1 of Bar et al. (2010). In fact, we follow the same motivation of model formulation as reviewed in Bar et al. (2010): modeling the error variance as random induces shrinkage and combines information across genes, as does modeling the primary effect as random. As the factor analysis component remains in each model simplification of RRmix, the EM algorithm can be used to estimate model parameters, treating the hidden factors as missing data. The formulation of the FF, FR, RF, and RR models are outlined next. Details of model specification and of the EM algorithm for each model are in Appendix B.

| *Model* | *Primary Effect* | *Error Variance* |
|---|---|---|
| RRmix | **R**andom (with **mix**ture structure) | **R**andom gene-specific |
| FF | **F**ixed | **F**ixed gene-specific |
| FR | **F**ixed | **R**andom gene-specific |
| RF | **R**andom | **F**ixed gene-specific |
| RR | **R**andom | **R**andom gene-specific |

Using each of these models, the goal is to estimate $\beta_{1g}$ and obtain p-value for significance of $\beta_{1g}$ for each gene $g$ with the model applied to a given SNP. In the context of eQTL, the fitting procedure for each model is similar to that for RRmix. We repeatedly fit the same model, each time with a different SNP, until we have $G \times N$ p-values, $N$ being the number of SNPs. For any given SNP, gene $g$ is classified as null or non-null with respect to the SNP based on the p-value for $\beta_{1g}$ which represents the SNP effect on gene $g$.

**FF and FR models**

For any gene $g$, let $\underset{n\times(2+r)}{X_a} = [X, X_c]$ and $\underset{(2+r)\times 1}{\eta_g} = \begin{bmatrix} \beta_g \\ \gamma_g \end{bmatrix}$, where $X$ and $X_c$ are as defined in the RRmix model (4.1). In the FF and FR models, the SNP effect $\beta_g$ is treated as a fixed effect, and error variance is treated as fixed in FF but assigned an Inverse-Gamma prior distribution with unknown hyperparameters in FR.

$$y_g = \mu + X_a\eta_g + \Lambda F_g + W_g$$

$$W_g \sim N_n(0, \sigma_g^2 I_n), \qquad F_g \sim N_q(0, I_q).$$

#### RF and RR models

In the RF and RR models, the SNP effect $\beta_g$ is treated as random effect and assigned a normal prior with a diagonal covariance matrix. The error variance is treated as fixed in RF but assigned an Inverse-Gamma prior distribution with unknown hyperparameters in RR.

$$y_g = \mu + X\beta_g + X_c\gamma_g + \Lambda F_g + W_g$$

$$W_g \sim N_n(0, \sigma_g^2 I_n), \qquad F_g \sim N_q(0, I_q),$$

$$\beta_g \sim N_2\left( 0 \, , \begin{bmatrix} \sigma_0^2 & 0 \\ 0 & \sigma_1^2 \end{bmatrix} \right).$$

### 4.2.3 Practical considerations

#### Number of hidden factors

The choice of the number of hidden factors, denoted by $q$, is influential to methods in which unwanted variation is modeled via factor analysis. Gagnon-Bartsch and Speed (2012) applied their RUV-2 method with different values of $q$, and examined quality assessment results for differential expression analysis. They concluded that the number of correct discoveries of their method increases with increasing number of hidden factors, but after a certain point, increasing number of hidden factors degrades performance. Gagnon-Bartsch et al. (2013) comment that the optimal way to choose $q$ is to run the analysis for several values of $q$ and choose the best one based on quality assessments, the outcome of which is influenced by human judgment.

We conjecture that the pattern of how number of hidden factors affects detec-

tion accuracy observed in Gagnon-Bartsch and Speed (2012) is applicable to our approach. Increasing the number of hidden factors before the range of most adequate number of hidden factors is reached results in additional hidden effects being accounted for. In contrast assuming more hidden factors than necessary causes the overall hidden effects to be over-estimated in the sense that variation due to the primary effect is incorrectly captured by the hidden factors, resulting in a loss in power.

For our family of models, the number of hidden factors $q$ is chosen based on the method described in Gao et al. (2014). Specifically, we perform principal component analysis (PCA) on the correlation matrix of the gene expression data and select the number of factors based on how many principal components (PCs) are required to explain the majority of the total variation as seen in a scree plot. Although visual examination of the scree plot is somewhat subjective, it provides guidance on the value of $q$, since the number of hidden factors that explain a part of the total variation, should not be very different to the number of PCs which explain the majority of total variation of gene expression data. Selecting a value of $q$ that is too large can result in negative eigenvalues when initializing $\Lambda$ via spectral decomposition. Thus, examination of the scree plot and the spectral decomposition scheme of initializing $\Lambda$ are helpful in pointing us toward the correct range of dimensionality of the hidden factor space. Within that range of values of $q$, the qualitative results of our methods appear to be insensitive to the value of $q$ , as we verify empirically in the simulation study.

**Unnecessity of modeling common hidden effects for all SNPs**

The model is fitted for all genes and for a single SNP at a time. The way the term $\Lambda\mathbf{F}$ is specified in the model may lead one to consider that the factors and loadings depend on the gene expression variables only, because $\Lambda\mathbf{F}$ adjusts the gene expression data to account for unwanted variation. However, while $\Lambda$ and $\mathbf{F}$ are referred to as loadings and factors in factor analysis terms, $\Lambda$ can be alternatively viewed, in terms of a regression model, as the matrix of $r$ hidden variables, each of length $n$, and $\mathbf{F}$ correspondingly can be viewed as the matrix of gene-level weights/effects of these hidden variables. Therefore, $\Lambda\mathbf{F}$ serves the purpose of accounting for "third party" hidden variables underlying the unwanted variation in light of the given SNP and the genes in the context of eQTL analysis. Such unmeasured variables exhibit non-zero correlations with genes and possibly non-zero correlation with the given SNP. That is, hidden factors may depend on the given SNP as well as the genes, and whether or not a hidden factor is orthogonal to a SNP is unverifiable in practice. This is why we allow the hidden effects $\Lambda\mathbf{F}$ to be heterogeneous across SNPs.

Additionally, the focus of analysis is not on learning the factors or their loadings but on estimating $\Lambda\mathbf{F}$ as a whole to enhance accuracy in estimation of the SNP effect $\boldsymbol{\beta}$ on the genes. As pointed out in Leek and Storey (2007), knowledge of $\Lambda\mathbf{F}$ suffices as long as the set of vectors identified as hidden factors spans the same linear space as the true set of hidden factors. One can even choose a set of convenient vectors, e.g. mutually orthogonal vectors, to be hidden factors, which is the approach used in Leek and Storey (2007). It is, therefore, unnecessary to restrict the factors or the loadings to be the same for all SNPs, as long as the computed $\Lambda\mathbf{F}$ captures total hidden effects that interfere with identification

of significant associations between each gene and the given SNP. Furthermore, even though factor loading matrix $\Lambda$ is only identifiable up to orthogonal rotations, a well-known fact in factor analysis, we verify below that $\Lambda\mathbf{F}$ estimated in RRmix is invariant under orthogonal rotation of $\Lambda$. Similar arguments can be made for each model in the family.

Let $\hat{\Lambda}$ be the estimate of $\Lambda$ in RRmix. We see from derivation (Appendix B.1) of RRmix estimates that for each gene $g$, the estimated hidden factor is $\hat{F}_g = \hat{b}_g \hat{F}_g^{(1)} + \left(1 - \hat{b}_g\right) \hat{F}_g^{(0)}$ in which $\hat{F}_g^{(1)}$ and $\hat{F}_g^{(0)}$ are directly estimated in the algorithm. Then, it can be seen from update rules of the EM algorithm of RRmix that an estimate of $\Lambda$ $\hat{\hat{\Lambda}} = \hat{\Lambda}P$ where $PP^T = I_n$ leads to estimates $\hat{\hat{F}}_g^{(1)} = P^T \hat{F}_g^{(1)}$ and $\hat{\hat{F}}_g^{(0)} = P^T \hat{F}_g^{(0)}$, which in turn leads to $\hat{\hat{F}}_g = P^T \hat{F}_g$. Therefore for each gene $g$, $\hat{\hat{\Lambda}}\hat{\hat{F}}_g = \hat{\Lambda}PP^T\hat{F}_g = \hat{\Lambda}\hat{F}_g$.

On the other hand, assuming hidden factors or loadings are the same across all SNPs may lead to the benefit of strength borrowing across SNPs in parameter estimation. However, model fitting in that case for one SNP is no longer independent of model fitting for another SNP, and computations are potentially too complex to be feasible.

## 4.3 Connections with HEFT

### 4.3.1 Connections between the family of models and HEFT

RRmix is applied repeatedly, each time for a different SNP, the same way as HEFT (Gao et al., 2014) for eQTL analysis. HEFT is implemented on standard-

ized data, i.e. each column of known data is centered and scaled to have variance 1.

Gao et al. (2014) observed that assuming a normal prior on $\beta_g$, i.e. $\beta_g \sim N(0, \Theta)$, is equivalent to imposing a ridge penalty for $\beta_g$. Rather than estimating $\Theta$, they decided to use $\Theta = I_2$.

If represented in a gene-specific framework, the HEFT model formulation is the same as RRmix apart from the following: i) the error variance is modeled as homogeneous across genes and fixed; ii) the prior distribution of the SNP effect is multivariate normal with identity covariance matrix; and iii) the covariate effect is treated as a random effect. That is, with gene expression $y_g^*$ for gene $g$, SNP vector $\mathbf{x}^*$, and covariate matrix $X_c^*$ representing standardized versions of $y_g$, $\mathbf{x}$, and $X_c$, and $X^* = [\mathbf{1}_n, \mathbf{x}^*]$, the HEFT model is

$$\begin{aligned} y_g^* &= \mu^* + X^*\beta_g^* + X_c^*\gamma_g^* + \Lambda^* F_g^* + W_g^* \\ W_g^* &\sim N_n(0, \sigma^2 I_n), \qquad F_g^* \sim N_q(0, I_q) \\ \beta_g^* &\sim N_2(0, I_2), \qquad \gamma_g^* \sim N_r(0, I_r), \end{aligned} \tag{4.8}$$

where effects of $r$ covariates are explicitly included. The differences i), ii), and iii) appear to be constraints that make the HEFT model appropriate for standardized data.

While implementation of our family of models strictly follows the EM algorithm, HEFT adopts an iterative algorithm similar to EM that produces approximate maximum likelihood parameter estimates. Following the HEFT model specifications, we illustrate the difference between EM algorithm updates rules and those in the HEFT algorithm in the next section.

The estimate of $\mu^*$ in HEFT is computed once via $\mu^* = \frac{1}{G}\sum_{g=1}^{G} y_g^*$ prior to

the start of the iterative scheme and is not further updated, and the response is normalized to $y_g^{**} = y_g^* - \mu^*$ for each gene in all subsequent estimations. In our family of models, $\mu$ corresponding to the original and unstandardized $y_g$ is updated in the M-step in each iteration.

Apart from RRmix, which does not require p-values for classification, inference about $\beta_{1g}$ is made via p-values derived from t-statistics in both HEFT and our family of models. However, it appears that the variance of the $\beta_{1g}$ estimate in the HEFT t-statistic is calculated in an unorthodox fashion involving the sample variance of residuals. For the FF and FR models, we obtain standard errors for the $\beta_{1g}$ estimates by calculating an approximation to the expected information matrix, accounting for the variance of missing data in an EM algorithm following Oakes (1999). For the RF and RR models we use the posterior covariance derived in the E-step as illustrated in Section 17.5.2 in Xu (2013) to form posterior t-statistics.

### 4.3.2 Modifications from EM algorithm in HEFT

We describe differences between the EM algorithm update rules based on the HEFT model and update rules of the HEFT algorithm step by step. For simplicity of exposition, we do not consider whether the data the algorithm is implemented on is standardized or not.

The HEFT model is equivalent to

$$H = \Omega\Gamma + W$$

where $H = Y - \mu\mathbf{1}_G^T$ is the $n \times G$ response matrix adjusted by $\mu$, $\Gamma$ is the $(2+r+q)\times G$ matrix of all effects (primary, covariate, and hidden), and $W$ is the $n \times G$ random

matrix of errors. Each matrix is partitioned into $G$ columns as follows,

$$H = [h_1, \ldots, h_G], \quad \Gamma = [\gamma_1, \ldots, \gamma_G], \quad W = [w_1, \ldots, w_G].$$

Thus the $g$th column of $H$ is given by

$$h_g = \Omega\gamma_g + w_g$$

Here $\Omega = [X, X_c, \Lambda]$. Assume $\gamma_g \sim$ i.i.d. $N(0, I_{2+r+q})$ independently of $w_g \sim$ i.i.d. $N(0, \sigma^2 I_n)$. The parameters to be estimated are $\Theta = \{\Lambda, \sigma^2\}$, and the missing data is $\Gamma = \{\gamma_g\}$.

The distributional assumptions imply

$$\begin{pmatrix} h_g \\ \gamma_g \end{pmatrix} \sim \text{i.i.d. } N\left[\begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} \Omega\Omega^T + \sigma^2 I_n & \Omega \\ \Omega^T & I_{2+r+q} \end{pmatrix}\right].$$

The complete data log-likelihood based on $H$ and $\Gamma$ written in the factorized form is, apart from an additive constant,

$$\begin{aligned} l_C &= \sum_{g=1}^{G} \Big( l(h_g|\gamma_g) + l(\gamma_g) \Big) \\ &= -\frac{G}{2} \log|\sigma^2 I_n| - \frac{1}{2\sigma^2} \sum_{g=1}^{G} (h_g - \Omega\gamma_g)^T (h_g - \Omega\gamma_g) \\ &\quad -\frac{G}{2} \log|I_{2+r+q}| - \frac{1}{2} \sum_{g=1}^{G} \gamma_g^T \gamma_g \\ &= -\frac{1}{2} \sum_{g=1}^{G} \gamma_g^T \left( I + \frac{1}{\sigma^2} \Omega^T \Omega \right) \gamma_g + \frac{1}{\sigma^2} \sum_{g=1}^{G} h_g^T \Omega \gamma_g \\ &\quad -\frac{Gn}{2} \log \sigma^2 - \frac{1}{2\sigma^2} \sum_{g=1}^{G} h_g^T h_g \,. \end{aligned} \tag{4.9}$$

**E-step**

The conditional mean and variance of $\gamma_g$ given $h_g$ are

$$\begin{aligned} \mathbb{E}(\gamma_g|h_g) &= \Omega^T (\Omega\Omega^T + \sigma^2 I_n)^{-1} h_g \\ &= \frac{1}{\sigma^2} \left( I_{2+r+q} + \frac{1}{\sigma^2} \Omega^T \Omega \right)^{-1} \Omega^T h_g \end{aligned} \tag{4.10}$$

and

$$\begin{aligned}\mathbb{V}(\gamma_g|h_g) &= I_{2+r+q} - \Omega^T(\Omega\Omega^T + \sigma^2 I_n)^{-1}\Omega \\ &= \left(I_{2+r+q} + \frac{1}{\sigma^2}\Omega^T\Omega\right)^{-1}, \end{aligned} \tag{4.11}$$

where the last quantity follows from the Woodbury identity.

Let

$$Q(\Theta|\Theta^*) = E^*_{\Gamma|H}(l_C)$$

denote the expected complete data log-likelihood given the current value of $\Theta$, $\Theta^*$. Then

$$\begin{aligned}Q(\Theta|\Theta^*) &= -\frac{1}{2}\sum_{g=1}^{G}\mathrm{tr}\left[\left(I + \frac{1}{\sigma^2}\Omega^T\Omega\right)V^*\right] - \frac{1}{2}\sum_{g=1}^{G}\gamma_g^{*T}\left(I + \frac{1}{\sigma^2}\Omega^T\Omega\right)\gamma_g^* + \frac{1}{\sigma^2}\sum_{g=1}^{G}h_g^T\Omega\gamma_g^* \\ &\quad -\frac{Gn}{2}\log\sigma^2 - \frac{1}{2\sigma^2}\sum_{g=1}^{G}h_g^T h_g, \end{aligned} \tag{4.12}$$

where $\gamma_g^*$ and $V^*$ are (4.10) and (4.11) respectively evaluated at $\Theta^*$. The HEFT updates for $\gamma_g^*$ and $V^*$ shown in supplementary materials of Gao et al. (2014) agree with the EM updates.

**M-step**

If $\Omega$ were entirely unknown, differentiating $Q$ (4.12) with respect to $\Omega$ yields

$$\begin{aligned}\frac{\partial Q}{\partial \Omega} &= -\frac{1}{2\sigma^2}\frac{\partial}{\partial\Omega}\mathrm{tr}\left(G\Omega V^*\Omega^T + \Omega\Gamma^*\Gamma^{*T}\Omega^T - 2\Gamma^* H^T\Omega\right) \\ &= -\frac{1}{\sigma^2}\left[\Omega(GV^* + \Gamma^*\Gamma^{*T}) - H\Gamma^{*T}\right], \end{aligned} \tag{4.13}$$

using the fact that $\sum_{g=1}\gamma_g^{*T}\Omega^T\Omega\gamma_g^* = \mathrm{tr}(\Omega\Gamma^*\Gamma^{*T}\Omega^T)$ and $\sum_{g=1}h_g^{*T}\Omega\gamma_g^* = \mathrm{tr}(\Gamma^* H^T\Omega)$. Equating (4.13) to zero and solving for $\Omega$ results in

$$\Omega = H\Gamma^{*T}(M^*)^{-1}. \tag{4.14}$$

where $M^* = (G \times V^* + \Gamma^*\Gamma^{*T})$.

Since $\Omega$ is in fact partially unknown, i.e. $X$ is a known $n \times 2$ matrix and $X_c$ is

a known $n \times r$ matrix in $\Omega = [X, X_c, \Lambda]$, differentiating $Q$ with respect to $\Lambda$ by first writing $Q$ as a function of $\Lambda$ gives

$$\begin{aligned}
\frac{\partial Q}{\partial \Lambda} &= -\frac{1}{2\sigma^2}\frac{\partial}{\partial \Lambda}\mathrm{tr}\left(G\Lambda V_{22}^*\Lambda^T + \Lambda\Gamma_2^*\Gamma_2^{*T}\Lambda^T + 2\Gamma_2^*\Gamma_1^* X^T\Lambda + 2GV_{21}^* X^T\Lambda - 2\Gamma_2^* H^T\Lambda\right) \\
&= -\frac{1}{2\sigma^2}\frac{\partial}{\partial \Lambda}\mathrm{tr}\left(\Lambda M_{22}^*\Lambda^T + 2M_{21}^* X^T\Lambda - 2\Gamma_2^* H^T\Lambda\right) \\
&= -\frac{1}{\sigma^2}(\Lambda M_{22}^* + XM_{12}^* - H\Gamma_2^{*T}), \qquad (4.15)
\end{aligned}$$

where $M_{22}^*$ consists of the lower right $q \times q$ block of $M^*$, $M_{12}^*$ consists of the upper right $(2 + r) \times q$ block of $M^*$ (similar indexing applies to $V^*$), and $\Gamma_2^*$ consists of the last $q$ rows of $\Gamma^*$. Equating (4.15) to 0 and solving for $\Lambda$ results in

$$\Lambda = (H\Gamma_2^{*T} - XM_{12}^*)(M_{22}^*)^{-1} \qquad (4.16)$$

which is the update rule for $\Lambda$ when derivations follow the EM algorithm strictly.

The description of the HEFT algorithm in supplementary material of Gao et al. (2014) appears to imply that the update of $\Lambda$ is taken as the last $q$ columns of (4.14). In that case, writing $M^* = (M_{kl}^*)$ and $(M^*)^{-1} = (M_*^{kl})$, the estimate of $\Lambda$ is

$$\Lambda = H\Gamma^{*T}\begin{bmatrix} M_*^{12} \\ M_*^{22} \end{bmatrix} = \left(H\Gamma_2^{*T} - H\Gamma_1^{*T}(M_{11}^*)^{-1}M_{12}^*\right)M_*^{22}, \qquad (4.17)$$

since $M_*^{12} = -(M_{11}^*)^{-1}M_{12}^*M_*^{22}$ follows from expanding

$$M^*(M^*)^{-1} = \begin{bmatrix} M_{11}^* & M_{12}^* \\ M_{21}^* & M_{22}^* \end{bmatrix}\begin{bmatrix} M_*^{11} & M_*^{12} \\ M_*^{21} & M_*^{22} \end{bmatrix} = I$$

to equations

$$M_{11}^*M_*^{12} + M_{12}^*M_*^{22} = \mathbf{0} \qquad (4.18)$$

$$M_{21}^*M_*^{12} + M_{22}^*M_*^{22} = I_q\,. \qquad (4.19)$$

Furthermore, (4.19)$-M_{21}^*(M_{11}^*)^{-1}$(4.18) gives

$$M_*^{22} = \left(M_{22}^* - M_{21}^*(M_{11}^*)^{-1}M_{12}^*\right)^{-1}\,. \qquad (4.20)$$

Substituting (4.20) into (4.17) gives

$$\Lambda = \left(H\Gamma_2^{*T} - H\Gamma_1^{*T}(M_{11}^*)^{-1}M_{12}^*\right)\left(M_{22}^* - M_{21}^*(M_{11}^*)^{-1}M_{12}^*\right)^{-1}\,. \qquad (4.21)$$

which is different from the EM update of $\Lambda$ (4.16).

However, (4.21) is still not the actual update rule for $\Lambda$ used in the HEFT software. It appears that authors of HEFT observed that (4.14) is the estimate for $\Omega$ when it is entirely unknown and produced an estimate of $\Lambda$ mimicing the formula (4.14), which is

$$\Lambda = H\Gamma_2^{*T}(M_{22}^*)^{-1}, \tag{4.22}$$

in which

$$M_{22}^* = \begin{bmatrix}\mathbf{0} & I_q\end{bmatrix} M^* \begin{bmatrix}\mathbf{0} \\ I_q\end{bmatrix} = \begin{bmatrix}\mathbf{0} & I_q\end{bmatrix} \left(G \times V^* + \Gamma^*\Gamma^{*T}\right) \begin{bmatrix}\mathbf{0} \\ I_q\end{bmatrix} = G \times V_{22}^* + \Gamma_2^*\Gamma_2^{*T},$$

since $\left[\mathbf{0}, I_q\right] A \left[\mathbf{0}, I_q\right]^T$ returns the lower right $q \times q$ block for any $k \times k$ ($k > q$) matrix A. Therefore, the HEFT update of $\Lambda$ (4.22) is different to the EM update of $\Lambda$ (4.16), and is (4.16) augmented by a matrix $XM_{12}^*(M_{22}^*)^{-1}$.

The EM update for $\sigma^2$ is obtained by setting the derivative of $Q$ (4.12)

$$\frac{\partial Q}{\partial \sigma^2} = -\frac{1}{2\sigma^4}\mathrm{tr}\left(G\Omega V^*\Omega^T + \Omega\Gamma^*\Gamma^{*T}\Omega^T - 2\Gamma^* H^T\Omega + H^T H\right) - \frac{Gn}{2\sigma^2} \tag{4.23}$$

to zero giving

$$\sigma^2 = \frac{1}{Gn}\mathrm{tr}\left[(H - \Omega\Gamma^*)^T(H - \Omega\Gamma^*)\right] + \frac{1}{n}\mathrm{tr}(\Omega V^*\Omega^T). \tag{4.24}$$

The update in HEFT for $\sigma^2$ is

$$\sigma^2 = \frac{1}{Gn}\mathrm{tr}\left[(H - \Omega\Gamma^*)^T H\right]. \tag{4.25}$$

Difference between the EM update (4.24) and the HEFT update (4.25) is

$$\begin{aligned}
&\left\{\frac{1}{Gn}\mathrm{tr}\left[(H - \Omega\Gamma^*)^T(H - \Omega\Gamma^*)\right] + \frac{1}{n}\mathrm{tr}(\Omega V^*\Omega^T)\right\} - \frac{1}{Gn}\mathrm{tr}\left[(H - \Omega\Gamma^*)^T H\right] \\
&= -\frac{1}{Gn}\mathrm{tr}\left[(H - \Omega\Gamma^*)^T\Omega\Gamma^*\right] + \frac{1}{n}\mathrm{tr}(\Omega V^*\Omega^T) \\
&= -\frac{1}{Gn}\mathrm{tr}\left[H^T\Omega\Gamma^*\right] + \frac{1}{Gn}\mathrm{tr}\left[\Omega\Gamma^*\Gamma^{*T}\Omega^T\right] + \frac{1}{n}\mathrm{tr}(\Omega V^*\Omega^T) \\
&= -\frac{1}{Gn}\mathrm{tr}\left[H^T\Omega\Gamma^*\right] + \frac{1}{Gn}\mathrm{tr}\left[\Omega M^*\Omega^T\right],
\end{aligned} \tag{4.26}$$

which equals 0 only if $\Omega$ is entirely unknown. This is because, in that case (4.14)

holds and implies $\Omega M^* = H\Gamma^{*T}$ which, when substituted into (4.26), gives

$$\begin{aligned} -\frac{1}{Gn}\mathrm{tr}\left[H^T\Omega\Gamma^*\right] + \frac{1}{Gn}\mathrm{tr}\left[H\Gamma^{*T}\Omega^T\right] &= -\frac{1}{Gn}\mathrm{tr}\left[H^T\Omega\Gamma^*\right] + \frac{1}{Gn}\mathrm{tr}\left[\left(H\Gamma^{*T}\Omega^T\right)^T\right] \\ &= -\frac{1}{Gn}\mathrm{tr}\left[H^T\Omega\Gamma^*\right] + \frac{1}{Gn}\mathrm{tr}\left[\Omega\Gamma^* H^T\right] \\ &= 0\,. \end{aligned}$$

It follows that the HEFT update for $\sigma^2$ is different from the EM update.

Thus, the HEFT algorithm is not strictly an EM algorithm. It is an iterative algorithm that includes update rules which are modifications of those in an EM algorithm.

CHAPTER 5

# MODEL-BASED CLASSIFICATION WITH SIMULTANEOUS ADJUSTMENT FOR UNWANTED VARIATION NUMERICAL ILLUSTRATIONS

## 5.1 Simulation Study

### 5.1.1 Synthetic data

We would like to compare performance of our family of models using synthetic data that reflect characteristics of real bioinformatics data. Ideally, the ground truth is known for performance comparison. However, to evaluate methods that account for unwanted variation for detection of non-null features, simulating data in which true non-null features are known and the nature and magnitude of unwanted variation resembles that in real data is challenging. To overcome this obstacle, researchers present alternative approaches for evaluating methods. For example, Gagnon-Bartsch and Speed (2012) compare performance of methods using microarray data in which true differentially expressed genes are known. Fusi et al. (2012) acknowledge that reliable ground truth information is generally not available in eQTL studies hence information from the empirical fit of their method to the real eQTL data is used in generating artificial data. Here, we made synthetic eQTL data based on the lung airway data in Gao et al. (2014): we subsetted and modified the gene expression data, simulated genotype data based on the modified gene expression data, and simulated hidden effects based on the simulated genotype data. An advantage of such "partially" synthetic data scheme is that characteristics of the real gene expression data are

preserved to some extent and sparsity and strength of signals can be controlled. Having control of the number of signals as well as how strong these signals are in the synthetic data allow methods to have a chance to differ in detection performance from each other by how well they pick up the entire spectrum of strong to weak signals.

A non-null SNP is defined to be significantly associated with at least one gene, whereas a null SNP is defined to be one that is not significantly associated with any gene. There are $n = 116$ samples, $G = 7,575$ genes, and $N = 191,959$ SNPs in the lung airway data set. The lung airway gene expression data matrix (116 by 7,575) and genotype data matrix (116 by 191,959) are denoted as $Y_{dat}$ and $X_{dat}$, respectively. The number of genes in the simulated data is set to $G = 1,000$. That is, we extract the first $G$ genes from $Y_{dat}$. A total of $N_{sim} = 1,000$ SNP vectors for the samples are simulated. We simulated 10 data sets for each of two cases (large sample size, $n = 116$, and small sample size, $n = 30$) in order to evaluate the performance advantage of methods that involve shrinkage estimation. Key steps of the simulation scheme are outlined below.

**Simulation of non-null SNPs**

*Step 1*: Randomly permute elements within each column of $Y_{dat}$. Take the top-left $n \times G$ block of matrix $Y_{dat}$ as $Y_{per}$. Permutation is done to remove correlation between the simulated non-null SNP and several genes.

*Step 2*: Randomly select a gene in $Y_{per}$, $y$. Create a SNP vector, $x$ by assigning values 0, 1, 2 to elements of $x$ according to ordered elements of $y$ and a randomly selected proportion from proportions of 0, 1, 2 values observed in $X_{dat}$. For

example, if the randomly selected proportion of 0, 1, 2 values is $(0.52, 0.35, 0.13)$, i.e. the proportion of 0, 1, 2 values in some SNP in $X_{dat}$ is $(0.52, 0.35, 0.13)$, and the 52nd and 87th percentiles of the $n$ elements of $y$ are $y_{52}$ and $y_{87}$, then the elements of $x$ corresponding to elements of $y$ that are less than $y_{52}$ are assigned the value 0, the elements of $x$ corresponding to elements of $y$ that are greater than or equal to $y_{52}$ and less than $y_{87}$ are assigned the value 1, and the rest are assigned the value 2.

*Step 3*: Randomly select $m$ ($m < n$) elements in $x$ and replace them with the first $m$ elements of $x$. This induces variability on the strength of SNP-gene association in the simulated data. For the large-sample case with $n = 116$, use $m = 50$; for the small-sample case with $n = 30$, use $m = 0$.

*Step 4*: Significant associations between the generated SNP vector and the gene vectors are uncovered by comparing $p$-values obtained from fitting a simple model with Bonferroni correction of the 0.05 significance level, $\frac{0.05}{G \times N_{sim}}$.The model is fitted with fixed effects only and does not account for hidden effects. It is specified as

$$Y_{per} = X_f \eta + \Sigma \tag{5.1}$$

where $X_f = [\mathbf{x}, gen, dis, smo]$ is an $n$ by 4 matrix, i.e. the model accounts for fixed effects of SNP and of covariates gender, disease status, and smoking status. These covariates are as described in the lung airway study in Gao et al. (2014).

*Step 5*: Repeat steps 2 to 4 $N_1$ times, where $N_1 = 50$.

Among the $N_1$ simulated SNPs, the number of actual non-null SNPs is expected to be less than $N_1$ due to the randomization scheme. Denote the number of actual non-null SNPs as $N_{11}$.

**Simulation of null SNPs**

*Step 1*: Generate a null SNP vector by randomly assigning 0, 1, 2 values to its $n$ elements according to a randomly selected proportion from proportions of 0, 1, 2 values observed in $X_{dat}$. For example, if the randomly selected proportion of 0, 1, 2 values is $(0.62, 0.32, 0.06)$, then this vector is used as probability weights for drawing 0, 1, 2 values with replacement to form the null SNP vector.

*Step 2*: With the generated SNP in the place of $x$ in $X_f$ in (5.1), fit model (5.1) and use the same Bonferroni threshold as above on the $G$ many p-values to confirm whether the SNP is null or non-null.

*Step 3*: Repeat steps 1 and 2 $N_0$ times to generate multiple null SNPs, where $N_0 = N_{sim} - N_1$.

Among the $N_0$ simulated SNPs, the number of actual null SNPs is expected to be less than $N_0$ due to the randomization scheme. Denote the number of actual null SNPs as $N_{00}$.

Thus, the number of final non-null SNPs, $N_1^*$, in the simulated genotype data satisfies $N_1^* = N_{11} + (N_0 - N_{00})$. The simulated $n \times N_{sim}$ genotype data matrix is organized so that final non-null SNPs take up the first $N_1^*$ columns.

**Addition of hidden effects**

We adopt the same framework for generating hidden factors and their effects as in Section S.2.1 of Supplementary Materials for Gao et al. (2014). However, we choose our own values in generating hidden factors. The purpose of this procedure is to generate hidden factors that are correlated with SNPs to various

degrees.

*Step 1*: Among the $N_{sim}$ SNPs, randomly select $r$ ($r < n$) SNPs, and perform PCA on the correlation matrix of the selected SNPs. We use $r = 100$ for the large-sample case and $r = 30$ for the small-sample case.

*Step 2*: Use scores on the first principal component to create a factor $f$ of 5 levels: Assign subjects that correspond to the $\frac{n}{5}$ largest scores on the first principal component to group 1, assign subjects that correspond to the next $\frac{n}{5}$ largest scores on the first principal component to group 2, and so on, so that a factor indicating which of the 5 groups each subject belongs to is created. The elements in the factor $f$ are $-\frac{1}{5}, -\frac{1}{10}, 0, \frac{1}{10}, \frac{1}{5}$ for groups 1 to 5.

*Step 3*: Steps 1 and 2 define the process of generating a non-orthogonal factor. The generated factor is referred to as a non-orthogonal factor because it induces an effect that is non-orthogonal to the effect of the $r$ SNPs. Repeat the process to generate $q_{1,sim} = 2$ non-orthogonal factors.

*Step 4*: An orthogonal SNP factor is generated by randomly assigning values $-\frac{1}{5}, -\frac{1}{10}, 0, \frac{1}{10}, \frac{1}{5}$ to its $n$ elements. Repeat the process to generate $q_{0,sim} = 2$ orthogonal factors. Hence the generated non-orthogonal and orthogonal factors in steps 3 and 4 make up the $n \times q_{sim}$ hidden factor matrix $\Lambda_{sim}$, where $q_{sim} = q_{1,sim} + q_{0,sim} = 4$. Although step 3 may not prevent the generated factor from being orthogonal by chance and similarly step 4 may not prevent the generated factor from being non-orthogonal by chance, these steps suffice in generating hidden factors that exhibit a range of correlations to SNPs.

*Step 5*: Draw elements of the $q_{sim}$ by $G$ factor effect matrix $F_{sim}$ from $N(0, 1)$. The total hidden effect is represented by the quantity $\Lambda_{sim} F_{sim}$.

*Step 6*: Create simulated gene expression data matrix $Y_{sim}$ by adding the hidden effects to $Y_{per}$, i.e. $Y_{sim} = Y_{per} + \Lambda_{sim} F_{sim}$.

The values $-\frac{1}{5}, -\frac{1}{10}, 0, \frac{1}{10}, \frac{1}{5}$ used in generating hidden factors are determined by visually examining that the distribution of between-gene correlations of the resulting $Y_{sim}$ resembles the distribution of between-gene correlations of $Y_{dat}$. A plot of distributions of between-gene correlations of $Y_{per}$ and $Y_{sim}$ in one of the large-sample ($n = 116$) simulated data set and $Y_{dat}$ is shown below.

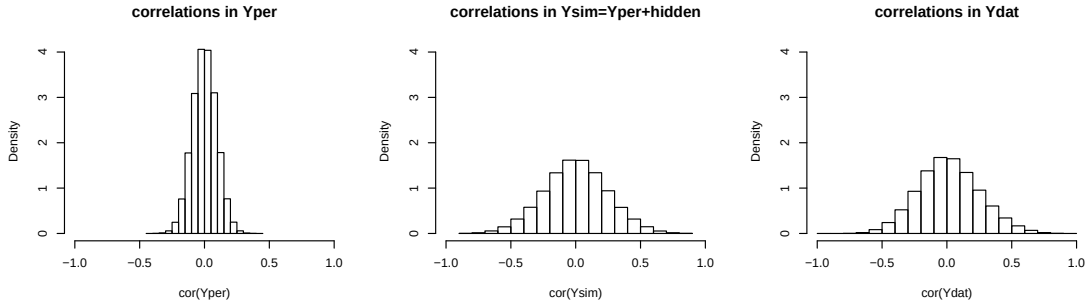


Figure 5.1: Distributions of between-gene correlations of $Y_{per}$ and $Y_{sim}$ in one of the large-sample simulated data and $Y_{dat}$ (left to right).

**Accounting for population structure and number of hidden factors in analysis**

Prior to conducting analysis, we performed PCA on the $n$ by $n$ covariance matrix of simulated genotype data and incorporated the first PC as a known covariate to account for population structure in simulated data, as suggested in Gao et al. (2014).

Also, we performed PCA on the $n$ by $n$ correlation matrix of simulated gene expression data $Y_{sim}$ and examined scree plot to determine the number of hidden factors, i.e. the value of $q$, that should be supplied for analysis. Hence, for the

large-sample case with $n = 116$, $q = 5$ was determined, and for the small-sample case with $n = 30$, $q = 3$ was determined. We found that using large values of $q$ such as $q = 7$ for the large-sample case and $q = 5$ for the small-sample case did not present qualitatively different analysis results (results not shown).

### 5.1.2 Evaluation of models

To evaluate performance in accounting for hidden effects, we implemented the family of models as well as a simple linear model and an R implementation of HEFT on simulated data sets generated in four settings based on the above simulation scheme: a) large-sample ($n = 116$) with added hidden effects (i.e. when $Y_{sim}$ is used as gene expression data), b) large-sample without added hidden effects (i.e. when $Y_{per}$ is used as gene expression data), c) small-sample ($n = 30$) with added hidden effects, and d) small-sample without added hidden effects. In Figure 5.3, boxplot of area under ROC curve (AUC) for a false positive rate cutoff of 0.05 and 0.001, respectively, of each model is shown. For each model, we calculated false discovery rate ($FDR = \frac{FP}{TP+FP}$) and power ($PWR = \frac{TP}{TP+FN}$), and created a boxplot over simulations (Figure 5.4) to tell a more complete story of performance than examining AUC alone. Implementation time of the models is compared in Figure 5.5.

As RRmix is a hierarchical mixture model for which detection of significant gene-SNP association is based on the posterior mean of the gene-specific latent indicator rather than p-values, accuracy measurements such as FDR and PWR of these models are not directly comparable with those of the other models. Nonetheless, we adopted an empirical threshold of 0.9999 for $b_g$ for the pro-

duction of FDR and PWR boxplots to reflect performance of RRmix. The empirical threshold was determined as follows: for each of 11 selected threshold value between 0.9 and 0.999999, FDR was calculated for RRmix classification on each simulated data set in the large-sample with added hidden effects case; because for each simulated data set there was a smallest threshold value that corresponded to FDR< 0.1, averaging the smallest threshold values across simulated data sets gave an empirical threshold value of approximately 0.9999.

LM is a simple linear regression model that includes SNP and covariate effects only and does not account for hidden effects. The publicly available HEFT application was written in C++ by authors of Gao et al. (2014). To compare all methods implemented in the same computing environment, we followed the HEFT model and algorithm as exactly as we can, implemented in R, and named the resulting replicate method HEFT-R. We verified that HEFT-R is a viable replicate of HEFT. The plotted t-statistic values, each for a gene-SNP association, produced by HEFT against those by HEFT-R, both with $q = 5$ assumed, for the sample data ($n = 116$, $G = 200$, $N = 200$) distributed in the HEFT software appears to hug the line $y = x$ tightly, especially towards t-statistics of large magnitudes (Figure 5.2). As detections are associated with t-statistics of large magnitudes, we verified that inferential results and in particular discoveries of HEFT-R resembled those of HEFT closely.

Compared to when no hidden effects were added in gene expression data, LM is associated with decreased AUC, increased FDR, and decreased power when hidden effects were added to gene expression data. This verifies that the added hidden effects mimicked realistic hidden effects in causing degradation of detection accuracy.

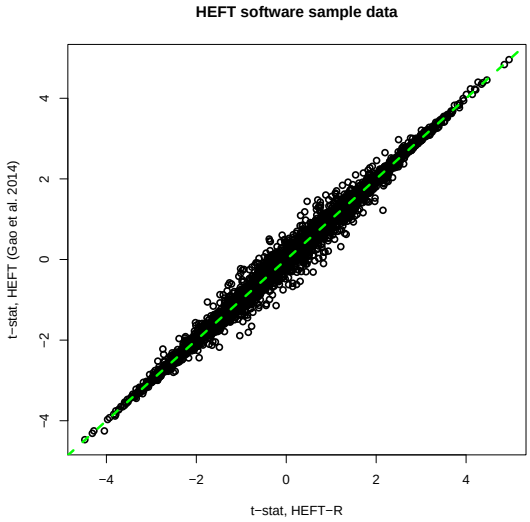


Figure 5.2: Comparison of t-statistic values produced by HEFT (Gao et al., 2014) (y-axis) to those by HEFT-R (x-axis) for the sample data distributed along with the HEFT software. The sample data contains 200 gene expression variables on 116 subjects, and 200 SNPs on the subjects. Number of hidden factors $q = 5$ was assumed for implementation of both methods. Most points and those at the largest magnitudes fall on the green dashed line of $y = x$.

The high values of area under ROC curve for RRmix in both the large-sample and small-sample cases suggest that RRmix is a good classifier. Though not directly comparable, FDR and PWR of RRmix are favorable given a threshold of 0.9999 for $b_g$. However, as expected RRmix was the slowest model to implement, due to the amount of gene-level missing data in the EM algorithm. Computational speed of RRmix may render the model unattractive when it is fitted on high-dimensional gene expression data repeatedly, each time for a different SNP, on real eQTL data.

Among models that were faster than RRmix in both the large-sample and small-sample cases, FF and FR are outstanding in their high AUC, low FDR and high PWR, and exhibit matching performance to HEFT-R. Compared to FF,

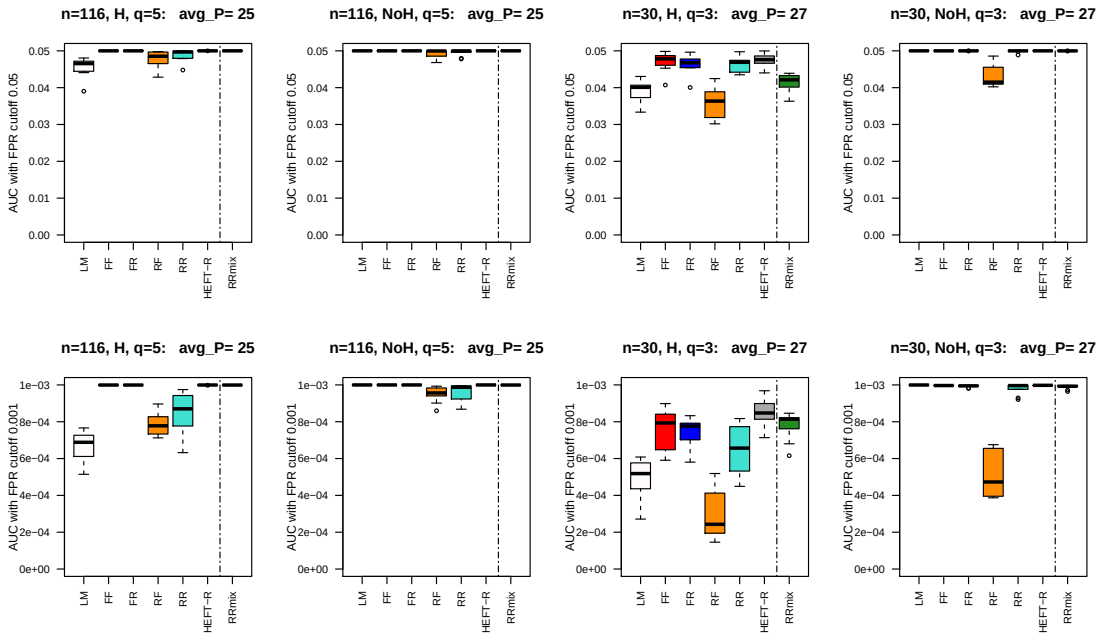


Figure 5.3: Boxplot of area under ROC curve for false positive rate cutoff of 0.05 (top row) and 0.001 (bottom row) for each model implemented on gene expression data with (columns 1 and 3) and without added hidden effects (columns 2 and 4) in the large-sample (columns 1 and 2) and small-sample (columns 3 and 4) cases. The average number of true gene-SNP associations in the ten simulated data sets is 25 in the large-sample case, and 27 in the small-sample case.

shrinkage estimation of gene-specific error variance in FR shows its advantage in lowering FDR when no hidden effects were added in the gene expression data. Such advantage is less visible in the large-sample case, because as sample size grows, the shrinkage in $\hat{\sigma}_g^2$ equation (4.4) becomes minimal.

RF and RR appear to be acceptable classifiers for simulated data with large samples, apart from a few high FDR values for RR in the large-sample with added hidden effects case. This is because they are associated with near 0 FDR but their PWR values are relatively low, which means that they made only a

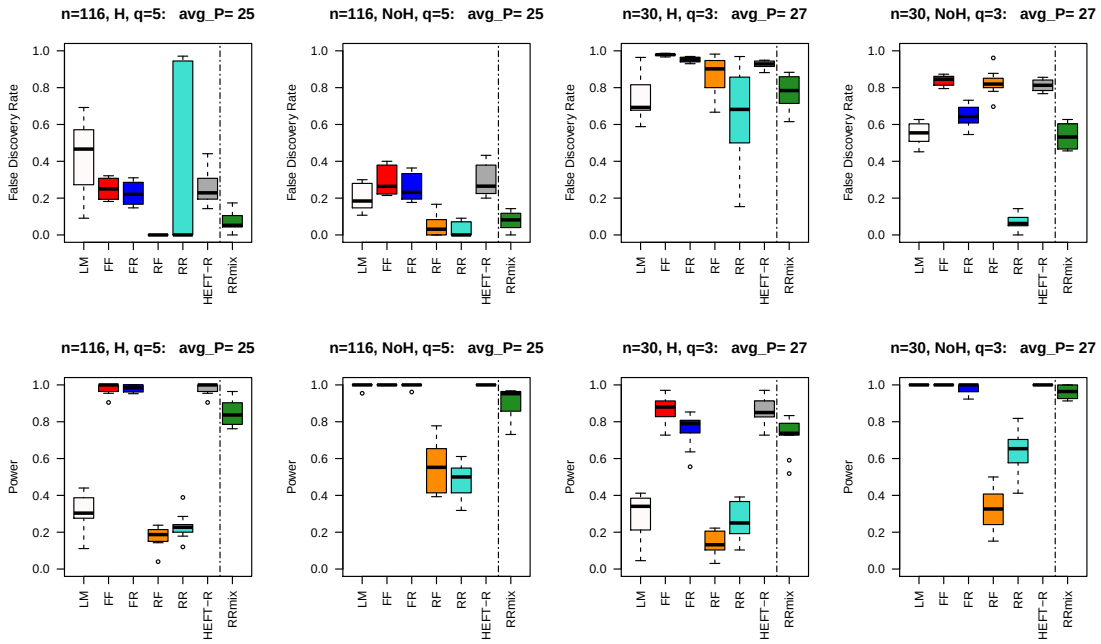


Figure 5.4: Boxplot of false discovery rate (top row) and power (bottom row) boxplot for each model implemented on gene expression data with (columns 1 and 3) and without added hidden effects (columns 2 and 4) in the large-sample (columns 1 and 2) and small-sample (columns 3 and 4) cases. The average number of true gene-SNP associations in the ten simulated data sets is 25 in the large-sample case, and 27 in the small-sample case.

few discoveries which were almost all true positives. For example, in the large-sample case with added hidden effects, RF has median power of about 0.2. Since the average number of true positives in the simulated data is 25, RF detected about $0.2 \times 25 = 5$ true positives. However, the near 0 FDR for RF in the same setting indicates that almost 100% of the discoveries by RF were true positives. Hence, we can deduce that RF only detected about 5 significant associations which were all true positives. RF and RR can be regarded as "conservative" classifiers that make few but mostly true discoveries. We investigate in Sections 5.1.3 and 5.1.4 possible causes for high FDR of RR in the large-sample with

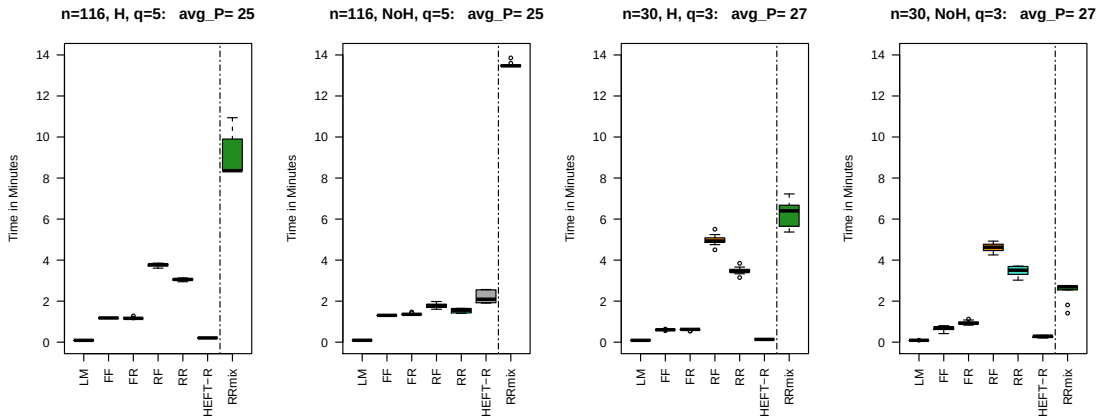


Figure 5.5: Boxplot of implementation time in minutes for each model implemented on gene expression data with (columns 1 and 3) and without added hidden effects (columns 2 and 4) in the large-sample (columns 1 and 2) and small-sample (columns 3 and 4) cases. The average number of true gene-SNP associations in the ten simulated data sets is 25 in the large-sample case, and 27 in the small-sample case.

added hidden effects case.

RR is associated with lower FDR and higher PWR than RF in the small-sample case, as well as generally higher AUC values than RF, due to shrinkage induced in both estimation of error variance and estimation of primary effect in RR. As the sample size grows, advantage of RR in PWR and FDR over RF becomes weaker, because of the same reason for the diminishing advantage in shrinkage estimation of error variance of FR over FF explained above. Comparing simulation results for no added hidden effects cases between large-sample and small-sample, we see that advantage of induced shrinkage by modeling the primary effect as random in RR is reflected in the difference in FDR between RR and FR, which is smaller in the large-sample case than the small-sample case. We show in Section 5.1.6 that inference on the primary effect based on RR

becomes closer to that based on FR as sample size increases.

### 5.1.3 Identifiability

We consider investigating identifiability for the model

$$y_g = \mu + X\beta_g + X_c\gamma_g + \Lambda F_g + W_g$$

$$W_g \sim N_n(0, \sigma_g^2 I_n)$$

$$F_g \sim N_q(0, I_q),$$

where $\underset{n\times 2}{X} = [\mathbf{1}_n, \mathbf{x}]$, $X_c$ is the matrix of covariates, and $\gamma_g$ is covariate effect, for gene $g$.

The marginal distribution of $y_g$ for each gene g is a multivariate normal distribution with covariance matrix denoted as $\Sigma_g$. If $\Sigma_g$ does not identify free parameters uniquely, there would be an identifiability issue. It was initially thought this might be the reason behind the big differences in FDR observed for RR when RR is implemented on simulated data in the large-sample with added hidden effects case.

First, we check whether the marginal covariance matrix of $y_g$ identifies parameters uniquely for FF and FR models, as they are simpler models and involve fewer parameters than RF and RR. Then, we construct further arguments for RF and RR building upon arguments developed for FF and FR. Owing to similarities between our models and HEFT (Gao et al., 2014), we also look into identifiability of HEFT to further understand its feasibility. We conclude that for all of the above models, the marginal covariance of response data identifies parameters uniquely, and hence that identifiability is not a likely reason for the high FDR in RR.

**FF and FR**

Here $\beta_g = \begin{bmatrix} \beta_{0g} \\ \beta_{1g} \end{bmatrix}$ is treated as a fixed effect. Covariance matrix of marginal distribution of $y_g$ is

$$\Sigma_g = \Lambda\Lambda^T + \sigma_g^2 I_n \,.$$

Factor loading $\Lambda$ is identifiable only up to orthogonal rotations. That is, for $\Lambda^* = \Lambda P$ where $P$ is a $q \times q$ orthogonal matrix, $\Lambda^*\Lambda^{*T} = (\Lambda P)(P^T\Lambda^T) = \Lambda\Lambda^T$. However, it does not pose a problem as we are interested in the estimate of the entire $\Lambda F$ which captures overall hidden effects, as explained in Section 4.2.3. For simplicity in exposition, we use $M$ to represent the $n \times n$ symmetric matrix $\Lambda\Lambda^T$ with $rank(M) \leq q$, and we use $M$ directly to denote the distinct elements in $M$ when parameters in the model are considered. Let $\boldsymbol{\sigma}^2$ denote the vector of $\sigma_g^2$, $g = 1, \ldots, G$.

Suppose there are two distinct vectors $\boldsymbol{\theta}_1 = (M_1, \mathbf{d}_1)$ and $\boldsymbol{\theta}_2 = (M_2, \mathbf{d}_2)$, where $\mathbf{d}_1 = (d_{11}, \ldots, d_{1G})^T$ and $\mathbf{d}_2 = (d_{21}, \ldots, d_{2G})^T$, that unknown parameters $\{M, \boldsymbol{\sigma}^2\}$ can take to satisfy

$$\Sigma_g = M + \sigma_g^2 I_n \tag{5.2}$$

for each $g$. $\Sigma_g$ is fixed for each $g$ implies that

$$M_1 - M_2 = \left(d_{2g} - d_{1g}\right) I_n \tag{5.3}$$

for each $g$.

First, we show by contradiction that $\boldsymbol{\theta}_1 \neq \boldsymbol{\theta}_2$ if and only if $M_1 \neq M_2$ and $\mathbf{d}_1 \neq \mathbf{d}_2$. If $M_1 = M_2$ and $\mathbf{d}_1 \neq \mathbf{d}_2$, then $d_{1g} = d_{2g}$ for each $g$ follows from (5.3), which is a contradiction. Similarly, if $M_1 \neq M_2$ and $\mathbf{d}_1 = \mathbf{d}_2$, (5.3) implies $M_1 = M_2$, which is, again, a contradiction.

Next, we show that $M$ and $\boldsymbol{\sigma}^2$ are uniquely determined. Consider the $nG$ by

$n$ matrix $(\Sigma_1, \ldots, \Sigma_G)^T$. That distinct $\boldsymbol{\theta}_1$ and $\boldsymbol{\theta}_2$ both satisfy (5.2) implies that

$$\mathbf{1}_G \otimes (M_1 - M_2) = (\mathbf{d}_2 - \mathbf{d}_1) \otimes I_n \,.$$

However, considering the rank of the left-hand side (LHS) and that of the right-hand side (RHS), we see that

$$rank(LHS) = rank(\mathbf{1}_G) \times rank(M_1 - M_2) = 1 \times rank(M_1 - M_2) \leq 2q$$

$$rank(RHS) = rank(\mathbf{d}_2 - \mathbf{d}_1) \times rank(I_n) = 1 \times n = n \,,$$

by the product rule of rank of Kronecker products, and that $rank(A + B) \leq rank(A) + rank(B)$ for two matrices $A$ and $B$ of the same dimensions. This is a contradiction to $rank(LHS) = rank(RHS)$ (unless $q \geq n/2$, which is hardly the case in practice). Thus, $\Lambda\Lambda^T$ and $\boldsymbol{\sigma}^2$ are uniquely determined in the FF and FR models.

**RF and RR**

SNP effect $\beta_g$ in RF and RR are assigned the following prior

$$\beta_g \sim N_2\left( 0 \,, \begin{bmatrix} \sigma_0^2 & 0 \\ 0 & \sigma_1^2 \end{bmatrix} \right).$$

Covariance matrix of marginal distribution of $y_g$ is

$$\Sigma_g = X \begin{bmatrix} \sigma_0^2 & 0 \\ 0 & \sigma_1^2 \end{bmatrix} X^T + \Lambda\Lambda^T + \sigma_g^2 I_n$$

$$= \sigma_0^2 \mathbf{1}_n \mathbf{1}_n^T + \sigma_1^2 \mathbf{x}\mathbf{x}^T + \Lambda\Lambda^T + \sigma_g^2 I_n \,.$$

Again, as $\Lambda$ is identifiable only up to orthogonal rotations, we write $\Lambda\Lambda^T$ as $M$, an $n$ by $n$ symmetric matrix with $rank(M) \leq q$. Let

$$S = \sigma_0^2 \mathbf{1}_n \mathbf{1}_n^T + \sigma_1^2 \mathbf{x}\mathbf{x}^T + M \,.$$

Since $rank(S) \leq 2 + q$, by a similar argument to that for FF and FR models it can be shown that $S$ and $\boldsymbol{\sigma}^2$ are uniquely determined.

We show further that parameters in $S$ are uniquely determined. Let $\boldsymbol{\theta}_1 = (a_1, b_1, M_1)$ and $\boldsymbol{\theta}_2 = (a_2, b_2, M_2)$ be two values of $\boldsymbol{\theta} = (\sigma_0^2, \sigma_1^2, M)$ that are both solutions of $S$ when $S$ is regarded as a function of $\boldsymbol{\theta}$. $S(\boldsymbol{\theta}_1) = S(\boldsymbol{\theta}_2)$ implies

$$(a_1 - a_2)\,\mathbf{1}_n\mathbf{1}_n^T + (b_1 - b_2)\,\mathbf{x}\mathbf{x}^T = M_2 - M_1\,.$$

Either that $rank(LHS) \leq 2$ or that $rank(RHS) \leq q$ suggests that the $n$ by $n$ matrix $(a_1 - a_2)\,\mathbf{1}_n\mathbf{1}_n^T + (b_1 - b_2)\,\mathbf{x}\mathbf{x}^T$ is not of full rank and hence its determinant is 0.

To further simplify notations, we write $\delta_0 = a_1 - a_2$ and $\delta_1 = b_1 - b_2$. Then the $(i, j)$ element of $\delta_0\mathbf{1}_n\mathbf{1}_n^T + \delta_1\mathbf{x}\mathbf{x}^T$ is $\delta_0 + \delta_1 x_i x_j$, where $x_i$ is the $i$-th element of $\mathbf{x}$. For any column $j$, $j = 1, \ldots, n$, the determinant of matrix $\delta_0\mathbf{1}_n\mathbf{1}_n^T + \delta_1\mathbf{x}\mathbf{x}^T$ can be written in terms of minors $det\left(B_{i,j}\right)$, $i = 1, \ldots, n$, where $B_{i,j}$ is the $(n-1)$ by $(n-1)$ matrix that results from deleting the $i$-th row and $j$-th column of $\delta_0\mathbf{1}_n\mathbf{1}_n^T + \delta_1\mathbf{x}\mathbf{x}^T$. That is, for any column $j$, $j = 1, \ldots, n$,

$$\begin{aligned}
& 0 = det\left(\delta_0\mathbf{1}_n\mathbf{1}_n^T + \delta_1\mathbf{x}\mathbf{x}^T\right) \\
\Longrightarrow \qquad & 0 = \sum_{i=1}^{n}(-1)^{i+j}\left(\delta_0 + \delta_1 x_i x_j\right) det\left(B_{i,j}\right) \\
\Longrightarrow \qquad & 0 = \delta_0\sum_{i=1}^{n}(-1)^{i+j} det\left(B_{i,j}\right) + \delta_1 x_j\sum_{i=1}^{n}(-1)^{i+j} x_i det\left(B_{i,j}\right)\,. \qquad (5.4)
\end{aligned}$$

However, (5.4), which holds for all column $j$, $j = 1, \ldots, n$, holds for almost all $\mathbf{x}$ only when both coefficients $\delta_0$ and $\delta_1$ are equal to zero, which is only possible when $a_1 = a_2$ and $b_1 = b_2$. That $a_1 = a_2$ and $b_1 = b_2$ in turn implies $M_1 = M_2$. Therefore, $\boldsymbol{\theta}_1 = \boldsymbol{\theta}_2$, i.e. $\sigma_0^2$, $\sigma_1^2$, and $\Lambda\Lambda^T$ in $S$ are uniquely determined.

**HEFT**

The HEFT model is applied to standardized data. We write the model as

$$y_g^* = \mu^* + X^*\beta_g^* + X_c^*\gamma_g^* + \Lambda F_g^* + W_g^*$$

$$W_g^* \sim N_n(0, \sigma^{*2} I_n), \qquad F_g^* \sim N_q(0, I_q)$$

$$\beta_g^* \sim N_2(0, I_2), \qquad \gamma_g^* \sim N_r(0, I_r)\,,$$

explicitly including the covariate effect which is treated as a random effect in Gao et al. (2014). Also, known vector $\mathbf{x}^*$ and known matrix $X_c^*$ are standardized (i.e. each column is centered and scaled to have variance 1) versions of $\mathbf{x}$ and $X_c$, and $X^* = [\mathbf{1}_n, \mathbf{x}^*]$. Let $s_g^2 = \dfrac{\sum_i (y_{ig} - \bar{y}_{.g})^2}{n-1}$ be the sample variance of $y_g$. The standardized response for each gene $g$ for which the HEFT model was built is $y_g^* = \dfrac{1}{s_g}\left(y_g - \bar{y}_{.g}\mathbf{1}_n\right)$.

Since covariance matrix of marginal distribution of $y_g^*$ is

$$\Sigma_g^* = X^* X^{*T} + X_c{}^* X_c{}^{*T} + \Lambda\Lambda^T + \sigma^{*2} I_n$$

in which $X^* X^{*T} + X_c{}^* X_c{}^{*T}$ is a known matrix, that $\Lambda\Lambda^T$ and $\sigma^{*2}$ are uniquely determined follows an argument similar to that for FF and FR models.

### 5.1.4 Algorithmic issues

The high FDR values of RR reflected in Figure 5.4 occur in simulated data sets 5, 7, and 10 for the large-sample with added hidden effects case. FDR of RR for those data sets are 0.94, 0.95, and 0.97, respectively, while FDR of RR for the other seven simulated data sets in the same configuration are all zero. We check convergence of EM algorithm of RR for each simulated data set. In particular, because the RR model is fitted to all genes but one SNP at a time, there are 1,000

EM algorithm implementations to be reviewed.

We see that for each of data sets 5, 7, and 10, convergence is achieved in notably fewer iterations for a few SNPs than the remaining SNPs, and for those offending SNPs RR detects a substantial number of false non-null genes which become false gene-SNP associations contributing to the high FDR when RR is implemented for that data set. Take data set 5 as an example (Figure 5.6). There are 2 SNPs, namely SNP 479 and SNP 722, out of the 1,000 SNPs, that are associated with convergence in 4 iterations, whereas the other 998 SNPs are associated with convergence in between 8 and 14 iterations. In the EM algorithm of RR, convergence is monitored through the value of $Q$ function evaluated at current parameter estimates. The value of $Q$ function upon convergence for SNP 479 and SNP 722 are the lowest among all converged values of $Q$ function, each converged value corresponding to RR implementation for one of the 1,000 SNPs in the simulated data set.

As inference is made via the posterior t-statistic for the hypothesis $H_0 : \beta_{1g} = 0$, $g = 1, \ldots, G$, we look into what happens to the distribution of $\beta_{1g}$, the SNP effect, for the offending SNPs. Plotting the empirical distribution of $\beta_{1g}$ for $g = 1, \ldots, G$ based on the RR output for SNPs for which non-null genes are detected, we observe that the empirical distribution of $\beta_{1g}$ for the offending SNPs differ from that for SNPs for which true non-null genes have been discovered. For instance, when applied to simulated data set 5, RR detects significant genes for SNPs 8, 9, 13, 20, 21, 29, 30, 479, and 722. Except for SNPs 479 and 722, detected genes for the other SNPs are true positives. In Figure 5.7, it can be clearly seen that empirical densities of $\beta_{1g}$ for SNPs 479 and 722 are in fact much flatter than those for the other SNPs, and such different empirical distributions correspond

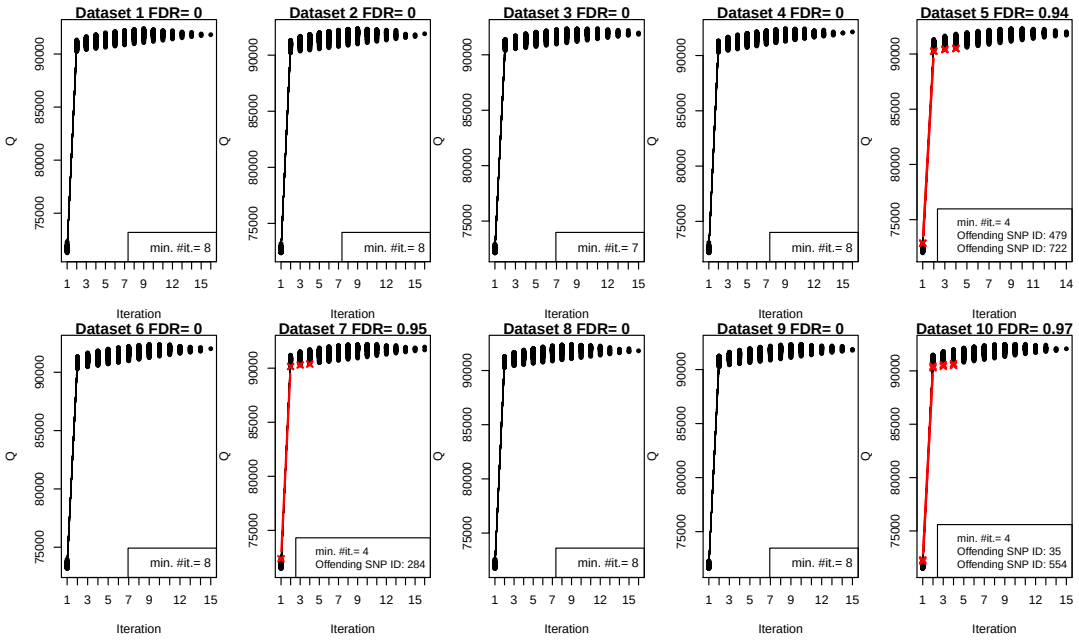


Figure 5.6: Plot of $Q$ function evaluated at current parameter estimates against iteration number corresponding to implementation of the EM algorithm of RR for each of the 1,000 SNPs in each of the 10 simulated data sets in the large-sample with added hidden effects configuration. For simulated data sets 5, 7, and 10, the $Q$ function versus Iteration for offending SNPs are labeled in red crosses, whereas all others are labeled as black filled circles. Minimum number of iterations and offending SNP IDs, if any, are shown in the bottom-right legend. In each grid that corresponds to a simulated data set, there is significant overlap of the black filled circles due to the fact that convergence of the EM algorithm for 1,000 SNPs are plotted.

to the many falsely detected genes.

Therefore, it seems that despite fast convergence, the short and low convergence curve for the few SNPs in the simulated data sets 5, 7, and 10 are caused by EM algorithm being trapped at some local maxima. As RR model involves estimation of $\sigma_0^2$ and $\sigma_1^2$ in the diagonal prior covariance matrix of $\beta_g$ whereas in FR $\beta_g$ is treated as a fixed effect, RR involves more parameters than FR. Hence,

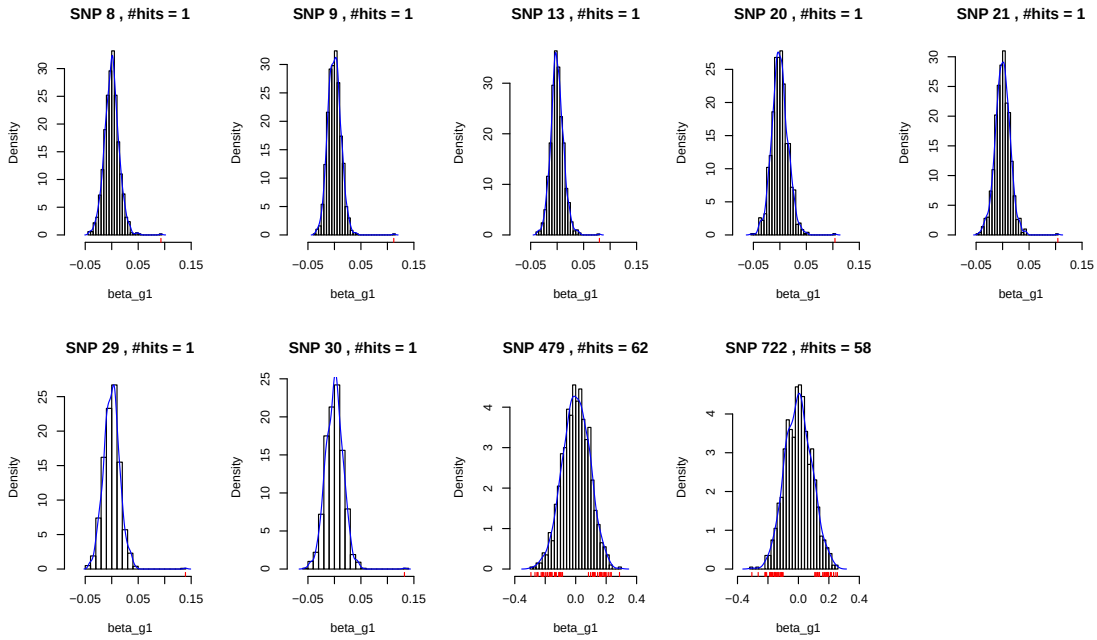


Figure 5.7: Empirical distribution of of $\beta_{1g}$, the SNP effect, for $g = 1, \ldots, G$ based on the estimates produced by the EM algorithm of RR, for SNPs for which non-null genes are detected in simulated data set 5 in the large-sample with added hidden effects case. Estimated density curve is added in blue. Red tick marks on the x-axis show the values of $\beta_{1g}$ for detected genes with respect to the given SNP. Number of detected hits by RR for each SNP is shown in the histogram title, along with the SNP ID. Total number of detected gene-SNP associations for the simulated data set is 127, majority of which is contributed by the offending SNPs 479 and 722.

there are more local maxima for the $Q$ function in RR and hence it is more likely for the EM algorithm to be trapped at a non-optimal one. Several strategies to overcome this problem are reviewed in Section 7.1 of Zhai (2008), which include exploring starting values and assigning further priors which act as constraints. However, running the EM algorithm for RR multiple times for exploration of starting values to avoid the algorithm being trapped at a local maxima is not realistic due to high-dimensionality of the data we analyze. A prior on $\sigma_1^2$ may

help constraining the search for solution in the parameter space, but it may introduce even more parameters into the model. On the other hand, in simulated data sets 5, 7, and 10, apart from the few offending SNPs, convergence of EM when implemented for the rest and majority of the SNPs appears to be reasonable. That is, RR performs sensibly for all but a few SNPs, and the algorithmic issue discussed is a likely reason behind the high FDR for RR.

Another possible reason we consider is that the estimate of $\sigma_1^2$ in the EM algorithm for RR may get so close to its boundary value 0 that the sequence of $\sigma_1^2$ estimates hardly move away from the boundary for several iterations before achieving convergence to a local maximizing value. In many problems involving constrained parameter space, if the maximizing parameter value is on the boundary of the parameter space, it may raise a concern for convergence of the EM algorithm (Nettleton, 1999). This would render the value of the $Q$ function converge slowly for several iterations before converging to notably higher values, and the EM algorithm is likely to be stopped by the error tolerance bound towards the beginning of those iterations of slow growth, resulting in the appearance of fast convergence of the algorithm which ends in a small number of iterations at less optimal parameter values. In our case, the true value of $\sigma_1^2$ is likely to be small, as we expect the non-null genes with respect to a given SNP to cluster closely, so the EM algorithm for RR may be susceptible to such an issue.

However, we rule out this possibility as we see that the estimate of $\sigma_1^2$ corresponding to offending SNPs are in fact the largest among $\sigma_1^2$ estimates corresponding to all 1,000 SNPs in the data set. For example, the estimate of $\sigma_1^2$ corresponding to SNP 479 and SNP 722 in data set 5 are around 0.01 whereas the estimate of $\sigma_1^2$ corresponding to the remaining SNPs lies between 0.000347

and 0.00155. Furthermore, we derive that the posterior t-statistic based on the EM algorithm update rules for RR is a function of $\sigma_1^2$ of the form $u(1/\sigma_1^2 + v)^{-1/2}$, where $u$ and $v$ are functions of the other parameters and observed data. Hence, under the assumption that the other parameters remain constant, the larger the estimate of $\sigma_1^2$ upon convergence of the EM algorithm, the larger the magnitude of the posterior t-statistic. This is consistent with the large number of false non-null genes with respect to the offending SNPs detected by RR. Therefore, the high FDR for RR is unlikely to be due to estimate of $\sigma_1^2$ in the EM algorithm getting too close to its boundary value.

### 5.1.5 Random or fixed error variance

It has been illustrated in the simulation study that shrinkage estimation of the gene-specific error variance is advantageous in lowering FDR, particularly in the small-sample case (Figure 5.4). However, such an advantage lessens as the sample size grows, because the amount of shrinkage in $\sigma_g^2$ (4.4) approaches 1, diminishing the impact of shrinkage regardless of whether there is added hidden effects in the data. That said, for applications where small samples are involved, models with shrinkage estimation of the gene-specific error variance are still preferable.

To illustrate the reduction in effect of shrinkage, we compare Figure 5.9 and Figure 5.8 for when added hidden effects are present in the simulated data. In each of the two figures, shrinkage of initial value for $\sigma_g^2$ from $m_g$ as indicated by expression (4.4) is confirmed, along with the strength of shrinkage becoming weaker as sample size becomes larger. Also, we note that effect of shrinkage on

the estimate of $\sigma_g^2$ produced by RR when compared with ML estimate produced by RF in Figure 5.9 for the large-sample case is less visible than in Figure 5.8 for the small-sample case; similar pattern can be observed for effect of shrinkage on the estimate of $\sigma_g^2$ produced by FR when compared with ML estimate produced by FF in these figures. Comparisons of the same kind for when there are no added hidden effects in the simulated data reveal similar findings (not shown).

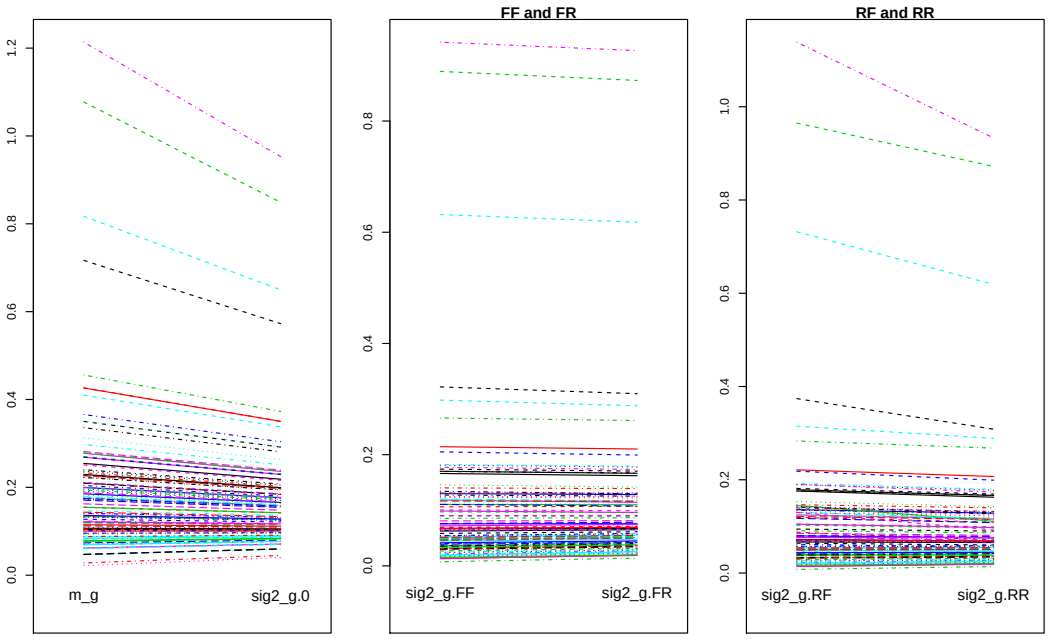


Figure 5.8: Comparison of $m_g$ computed on original response data and initial shrinkage estimate of $\sigma_g^2$ used as initial values for $\sigma_g^2$ for all models (left), comparison of estimate of $\sigma_g^2$ produced by FF and that by FR (middle), and comparison of estimate of $\sigma_g^2$ produced by RF and that by RR (right), for analysis on SNP 1 of simulated data set 1 in the *small-sample* with added hidden effects case. Values of $m_g$ and estimates of $\sigma_g^2$ are plotted on the y-axis, and only the first 100 genes are plotted for less condensed illustrations. Each line corresponds to one gene. Effect of shrinkage is reflected by the shrunken range of line end points on the right side of each plot.

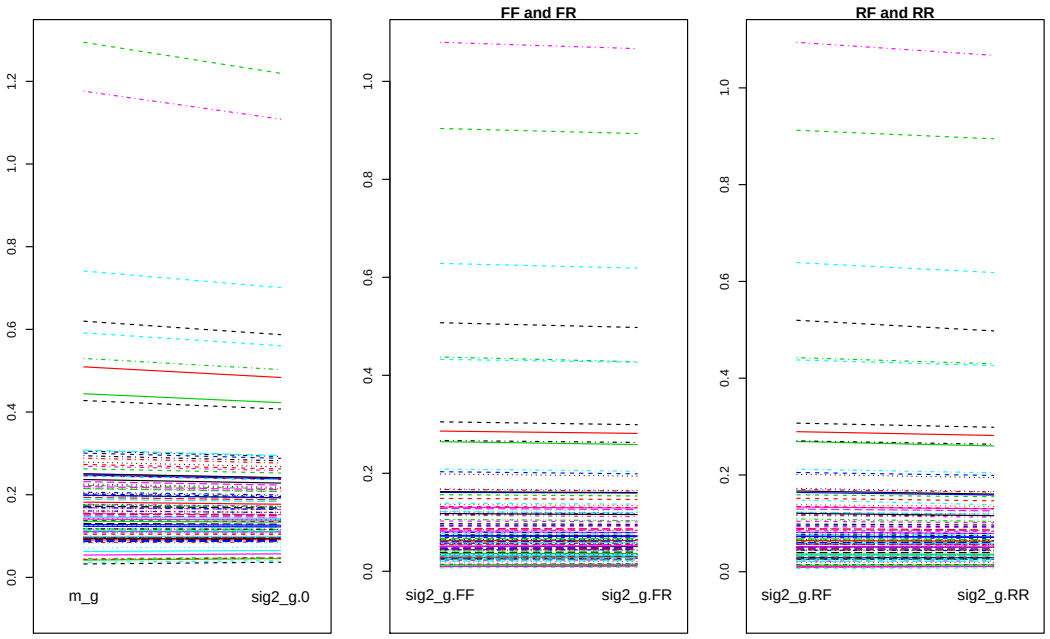


Figure 5.9: Comparison of $m_g$ computed on original response data and initial shrinkage estimate of $\sigma_g^2$ used as initial values for $\sigma_g^2$ for all models (left), comparison of estimate of $\sigma_g^2$ produced by FF and that by FR (middle), and comparison of estimate of $\sigma_g^2$ produced by RF and that by RR (right), for analysis on SNP 1 of simulated data set 1 in the *large-sample* with added hidden effects case. Values of $m_g$ and estimates of $\sigma_g^2$ are plotted on the y-axis, and only the first 100 genes are plotted for less condensed illustrations. Each line corresponds to one gene. Effect of shrinkage is reflected by the shrunken range of line end points on the right side of each plot.

It is observed in Figure 5.5 that models that involve shrinkage estimation of the gene-specific error variance differ in computational speed to their counterpart models where error variance is treated as a fixed effect. All models are implemented on the same computing system and the same error tolerance bound is used in all algorithms. RR is associated with shorter implementation time than RF in all cases of simulation - in fact, RF is the slowest to implement among

all models but RRmix in the family. Computational speed of FR is very similar to and at most marginally slower than that of FF.

### 5.1.6 Random or fixed primary effect

We illustrate that treating the primary effect as random in the RR model gives asymptotically equal values of t-statistics to when the primary effect is treated as a fixed effect in the FR model. In these models, the primary effect is the scalar quantity $\beta_{1g}$ that is in $\beta_g = \begin{bmatrix} \beta_{0g} \\ \beta_{1g} \end{bmatrix}$, the vector of effects for design matrix $\underset{n\times 2}{X} = [\mathbf{1}_n, \mathbf{x}]$. Suppose for each gene, the following quantities are known: sample means $\mu$, intercept $\beta_{0g}$, covariate effects $X_c\gamma_g$, hidden effects $\Lambda F_g$, and error variance $\sigma_g^2$. Let $h_g = y_g - \mu - \beta_{0g}\mathbf{1}_n - X_c\gamma_g - \Lambda F_g$. Model reduces to

$$h_g = \beta_{1g}\mathbf{x} + W_g$$

$$W_g \sim N_n(0, \sigma_g^2 I_n).$$

If $\beta_{1g}$ is treated as a random effect, suppose $\beta_{1g} \sim N(0, \sigma_1^2)$ with $\sigma_1^2$ known, then the joint distribution with observed data $h_g$ is

$$\begin{bmatrix} \beta_{1g} \\ h_g \end{bmatrix} \sim \quad N_{n+2}\left( 0\ , \quad \begin{bmatrix} \sigma_1^2 & \sigma_1^2\mathbf{x}^T \\ \mathbf{x}\sigma_1^2 & \mathbf{x}\sigma_1^2\mathbf{x}^T + \sigma_g^2 I_n \end{bmatrix} \right)$$

In terms of BLUP in a mixed effect model, mean $E(\beta_{1g}|h_g)$ and variance $var(\beta_{1g}|h_g)$ conditional on the observed data are

$$\begin{aligned} \tilde{\beta}_{1g} &= \sigma_1^2\mathbf{x}^T\left(\mathbf{x}\sigma_1^2\mathbf{x}^T + \sigma_g^2 I_n\right)^{-1} h_g \\ &= \left(\frac{\sigma_g^2}{\sigma_1^2} + \mathbf{x}^T\mathbf{x}\right)^{-1} \mathbf{x}^T h_g \end{aligned} \tag{5.5}$$

$$var\left(\tilde{\beta}_{1g}\right) = \sigma_1^2 - \sigma_1^2 \mathbf{x}^T \left(\mathbf{x}\sigma_1^2 \mathbf{x}^T + \sigma_g^2 I_n\right)^{-1} \mathbf{x}\sigma_1^2$$
$$= \sigma_1^2 - \left(\frac{\sigma_g^2}{\sigma_1^2} + \mathbf{x}^T\mathbf{x}\right)^{-1} \mathbf{x}^T\mathbf{x}\sigma_1^2$$
$$= \left(\frac{\sigma_g^2}{\sigma_1^2} + \mathbf{x}^T\mathbf{x}\right)^{-1} \sigma_g^2.$$

If $\beta_{1g}$ is treated as a fixed effect, maximum likelihood estimator (MLE) for the coefficient is

$$\hat{\beta}_{1g} = \left(\mathbf{x}^T\mathbf{x}\right)^{-1} \mathbf{x}^T h_g \tag{5.6}$$
$$var\left(\hat{\beta}_{1g}\right) = \left(\mathbf{x}^T\mathbf{x}\right)^{-1} \sigma_g^2.$$

It can be seen that (5.5) and (5.6) lead to

$$\tilde{\beta}_{1g} = \left(\frac{\sigma_g^2}{\sigma_1^2} + \mathbf{x}^T\mathbf{x}\right)^{-1} \left(\mathbf{x}^T\mathbf{x}\right)\hat{\beta}_{1g}$$
$$= \lambda_g \hat{\beta}_{1g},$$

where $\lambda_g = 1 - \dfrac{\sigma_g^2/\sigma_1^2}{\sigma_g^2/\sigma_1^2 + \mathbf{x}^T\mathbf{x}} = 1 - O(n^{-1})$. Hence for each $g$, $\tilde{\beta}_{1g}$ takes $\hat{\beta}_{1g}$ and shrinks towards 0, and $\tilde{\beta}_{1g} \approx \hat{\beta}_{1g}$ for large $n$. The t-statistic for BLUP and for MLE are, respectively,

$$t\left(\tilde{\beta}_{1g}\right) = \frac{\tilde{\beta}_{1g}}{\sqrt{\left(\dfrac{\sigma_g^2}{\sigma_1^2} + \mathbf{x}^T\mathbf{x}\right)^{-1} \sigma_g^2}} \qquad \text{and} \qquad t\left(\hat{\beta}_{1g}\right) = \frac{\hat{\beta}_{1g}}{\sqrt{\left(\mathbf{x}^T\mathbf{x}\right)^{-1} \sigma_g^2}}.$$

Their ratio approaches 1 as n grows, since

$$\frac{t\left(\tilde{\beta}_{1g}\right)}{t\left(\hat{\beta}_{1g}\right)} = \frac{\tilde{\beta}_{1g}}{\hat{\beta}_{1g}} \sqrt{\frac{\sigma_g^2/\sigma_1^2 + \mathbf{x}^T\mathbf{x}}{\mathbf{x}^T\mathbf{x}}} = \lambda_g \sqrt{\frac{1}{\lambda_g}} = \sqrt{\lambda_g} \to 1$$

as $n \to \infty$. Thus, for large $n$, inference on the primary effect $\beta_{1g}$ when it is treated as a random effect is approximately the same as when it is treated as a fixed effect, which means the advantage of induced shrinkage by treating the gene-specific main effect as random diminishes as the sample size becomes large. This explains the observation in Figure 5.4 that for simulated data with large sample size and no added hidden effects, the advantage of RR over FR in lowering FDR is not as clear as for simulated data with small sample size and no

added hidden effects.

Asymptotically equal inference on the primary effect between being treated as random and fixed effects implies that the FR and RR models have little inferential differences when sample size is large. In light of the algorithmic issues with RR and the fast computational speed of FR, we promote FR further as a fast alternative to RRmix for model-based classification in bioinformatics applications.

## 5.2 Applications

### 5.2.1 eQTL analysis with Lung Airway Data

#### Identification of eQTL

eQTL mapping is an area in genome-wide association studies (GWAS) that targets identification of genomic determinants that can be linked to complex diseases. eQTL mapping study became an extension of Quantitative Trait Loci (QTL) mapping study when thousands of gene expression abundances are made available via microarrays and such complex traits are considered simultaneously as phenotypes in mapping studies. High correlation between a genetic marker and the expression of a specific gene render the genetic marker an eQTL. Typical approaches for detection of eQTL have made use of linear regression models with gene expressions as response and markers as predictor, and statistical inference is drawn based on the estimation of marker effect parameters. The high-dimensional gene expression and marker data constitute the so-called "large $p$, small $n$" scenario, where the number of features $p$ is much larger than the sample size $n$. Moreover, because the number of true eQTL is dominated by the number of moderately correlated gene-marker pairs, eQTL detection is a case of sparse classification of features.

Unlike in QTL mapping where one or a few traits are considered as phenotypes, the large number of gene expressions lead to a complex phenotype covariance structure (Kendziorski et al., 2006). Furthermore, as summarized in Listgarten et al. (2010), major statistical challenges in eQTL come from two areas of statistical applications: GWAS and differential expression analysis. Con-

founding by population structure which eQTL inherits from GWAS must be addressed. Unwanted variation that confounds the biological variation of interest in the gene expression data, such as batch effects and other physical variables that correlate with batch effects (Gagnon-Bartsch and Speed, 2012), may present problems in inferential accuracy, as in differential expression analysis. This is because determining whether a genomic locus is associated with a gene expression level in eQTL is essentially checking if that gene is differentially expressed between individuals with different alleles at that locus. Statistical issues arise due to the high-dimensionality of data and they complicate identification and interpretation of eQTL (Michaelson et al., 2009).

Our model-based classification methods FF, FR and RRmix, are suitable for eQTL identification. A comparable method is HEFT (Gao et al., 2014).

We applied the algorithm based on the FR model on the same lung Small Airway Epithelium (SAE) data set used in Gao et al. (2014) to identify eQTL that affect gene expression. There are 79 smokers and 37 nonsmokers in the sample ($n = 116$). The gene expression data set contains expression level of 7,575 protein-coding genes for each individual, and 191,959 genotypes for each individual are available for analysis. To implement FR, we included the same covariates as in HEFT (Gao et al., 2014) in the model; namely gender, smoking status, and lung disease status. Following the treatment in Gao et al. (2014) to account for population structure, we computed the loading of the first factor from PCA on the genotype covariance matrix and include it as a fixed covariate. The number of hidden factors was set to be 5, which was selected by visually examining the eigen spectrum of the gene expression correlation matrix, same as in HEFT. We used a conservative Bonferroni corrected threshold of

$0.05/(7,575 \times 191,959) = 3.438578e\text{-}11$ for assessing significance of association in each gene-SNP pair.

A simple linear regression model (LM) that includes the same fixed covariates as FR was also implemented to provide baseline results. Table 5.1 shows the number of associations found by LM, HEFT, and our methods. LM discovered 134 significant gene-SNP associations, among which 80 are cis (i.e. the eQTL genotype is within 1 Mb upstream of the start position and 1 Mb of the end position of the coding region of the expressed gene with which it is associated), 65 are non-duplicated (i.e. there can be at most one significant cis association per overlapping gene region), and 39 are cis and non-duplicated. The 96 non-duplicated significant eQTL associations discovered by HEFT have 56 in common with the 134 hits discovered by LM. There are 61 cis and non-duplicated associations discovered by HEFT. Among the 39 cis and non-duplicated associations, 2 of them were identified by LM but not by HEFT. Manhattan and QQ plots on LM p-values for the 2 genes are shown in Figure 5.10. In fact, these 2 genes are associated with lung disease - preliminary results in Accordini et al. (2013) suggest that SPINK5 located on 5q32 or genes in linkage disequilibrium play a role in asthma severity in adults, and YBEY located on chromosome 21q22.3 is known to be associated with pneumonia and tuberculosis (the GeneCards encyclopedia, www.genecards.org). LM identified a cis association at SNP rs4529181 for SPINK5 and a cis association at SNP rs4818835 for YBEY.

It took about 33 hours via parallel R computing with the `snow` package on 7 nodes, each of which is with 16GB RAM and 8 processors, to complete the FR implementation on the lung SAE data. Among the 358 significant gene-SNP associations that FR identified, 217 are cis, 149 are non-duplicated, and 84 are

| | LM | | HEFT | | FR | RRmix |
|---|---|---|---|---|---|---|
| | | 56 | | 95 | | |
| significant | 134 | ↘ | | ↙ | 358 | 2,152 |
| cis | 80 | | | | 217 | 306 |
| non-duplicated | 65 | | 96 | | 149 | 358 |
| | | 2 | | 24 | | |
| cis and non-duplicated | 39 | ⟶ | 61 | ⟵ | 84 | 107 |
| Time | | | 13 hours (C++) | | 33 hours (R) | 430 hours (R) |

Table 5.1: Number of detected associations that were produced by LM, HEFT, FR, and RRmix. For LM and FR, detections were made with a Bonferroni corrected threshold on the p-values. Non-duplicated detections produced by HEFT were extracted from Gao et al. (2014) among which we verified 61 are cis. For RRmix, detections were made with a threshold of 0.9999 on the posterior mean of the gene-specific latent indicator.

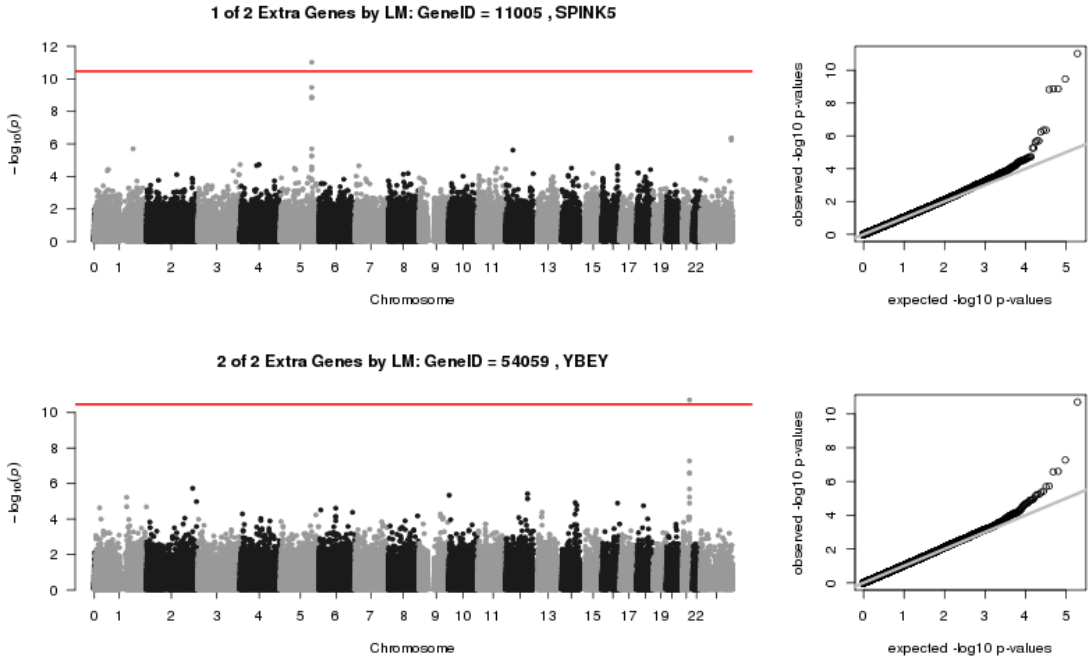


Figure 5.10: Manhattan and QQ plots for the two genes for which a significant cis association was identified by LM but not by HEFT. A reference line at $-\log_{10}(0.05/(7,575 \times 191,959)) = 10.46$ for the Bonforroni threshold is indicated in the Manhattan plots.

cis and non-duplicated eQTL associations. Compared to the 39 cis and non-duplicated associations discovered by LM, the 84 cis and non-duplicated associations discovered by FR includes all of them and 45 extra associations. In particular, the 2 cis and non-duplicated associations discovered by LM but not by HEFT were in fact detected by FR. Among the 96 non-duplicated gene-SNP associations discovered by HEFT, FR recovered 95 of them among the 358 hits and found more non-duplicated associations than HEFT, i.e. 149 compared to 96. The p-value of the one non-duplicated association discovered by HEFT, but not by FR, is equal to 2.583895$e$-11. It was ranked 94 out of the 96 hence fairly close to the Bonferroni corrected threshold $0.05/(7,575 \times 191,959) = 3.438578e$-11. The corresponding p-value produced by FR is 1.792242$e$-10, which is on the boundary of being declared significant. For that association, $-\log_{10}$ of the p-value produced by HEFT, the Bonferroni corrected threshold, and the p-value produced by FR are, respectively, 10.5877, 10.4636, 9.7467. Hence, the level of agreement of FR discoveries to HEFT discoveries is high.

Additionally, FR recovered all except 1 of the 61 cis and non-duplicated associations found by HEFT, 50 of which are at the exact same SNP as detected by HEFT. Table 5.3 shows p-values of the 24 cis and non-duplicated associations that were found by FR but not by HEFT. Manhattan and QQ plots of the 191,959 p-values for those 24 genes were produced. Some of those plots are included in Appendix C. The plots of top two of these genes, where ranking is based on ascending p-value, are shown in Figure 5.11. Therefore, it appears that FR is capable of accounting for hidden factors with matching performance to HEFT while detecting considerably more cis eQTL associations than HEFT for the lung SAE data.

| Ranking | Gene ID | Gene Symbol | SNP | HEFT p-value | FR p-value | Ref. on Lung Disease Association |
|---|---|---|---|---|---|---|
| 1 | 669 | BPGM | rs1160049 | 6.217475e-11 | 1.766539e-13 | Oparina et al. (2013) |
| 2 | 55234 | SMU1 | rs10758181 | 3.534760e-11 | 1.871564e-13 | |
| 3 | 7317 | UBA1 | rs3943122 | 4.750418e-09 | 2.274405e-13 | |
| 4 | 7174 | TPP2 | rs660236 | 2.831262e-10 | 2.302530e-13 | Golz et al. (2007) |
| 5 | 11005 | SPINK5 | rs4529181 | 8.935957e-11 | 2.970957e-13 | Accordini et al. (2013) |
| 6 | 286042 | FAM86B3P | rs2945886 | 3.624041e-11 | 4.840572e-13 | |
| 7 | 6716 | SRD5A2 | rs4952197 | 1.499098e-10 | 4.863430e-13 | |
| 8 | 28512 | NKIRAS1 | rs11922577 | 3.599428e-11 | 6.310906e-13 | Khodyrev et al. (2012) |
| 9 | 10950 | BTG3 | rs2824407 | 1.484254e-10 | 7.379107e-13 | Chen et al. (2013) |
| 10 | 23324 | MAN2B2 | rs7659958 | 6.835302e-11 | 7.468270e-13 | |
| 11 | 3839 | KPNA3 | rs9568313 | 4.396817e-11 | 1.274089e-12 | |
| 12 | 57134 | MAN1C1 | rs12075373 | 5.386916e-11 | 1.806777e-12 | Manichaikul et al. (2014) |
| 13 | 84300 | UQCC2 | rs594223 | 5.570096e-10 | 2.553291e-12 | |
| 14 | 127700 | OSCP1 | rs946680 | 1.205446e-10 | 3.007918e-12 | |
| 15 | 3128 | HLA-DRB6 | rs3763307 | 8.787947e-11 | 3.316236e-12 | |
| 16 | 54059 | YBEY | rs4818835 | 2.787466e-10 | 3.584022e-12 | the GeneCards encyclopedia (www.genecards.org) |
| 17 | 283078 | MKX | rs7358151 | 8.869921e-11 | 4.120812e-12 | |
| 18 | 8458 | TTF2 | rs12029623 | 2.351508e-10 | 4.835909e-12 | |
| 19 | 100131814 | LINC00271 | rs9389329 | 1.700244e-10 | 9.732215e-12 | |
| 20 | 80071 | CCDC15 | rs11219878 | 1.876345e-10 | 1.247935e-11 | Shin et al. (2012) |
| 21 | 26575 | RGS17 | rs1933260 | 7.620549e-10 | 2.096905e-11 | You et al. (2009) |
| 22 | 203427 | SLC25A43 | rs217994 | 5.602801e-10 | 2.463470e-11 | |
| 23 | 4552 | MTRR | rs179171 | 2.819637e-10 | 2.737549e-11 | |
| 24 | 23150 | FRMD4B | rs7623989 | 1.343265e-09 | 3.262657e-11 | |

Table 5.3: FR and HEFT p-values for each of 24 cis and non-duplicated associations that were found by FR but not by HEFT. Significant associations were determined by comparison to the Bonferroni corrected threshold $0.05/(7,575\times191,959) = 3.438578e\text{-}11$. HEFT p-values were extracted from data downloaded from the site at NCBI's Gene Expression Omnibus where authors of Gao et al. (2014) deposited their results. The last column shows references that indicate lung disease association of 9 genes.

Among cis associations discovered by FR but not by HEFT, genes in several of these cis associations have been shown in previous studies to be of relevance to lung disease. Besides SPINK5 (Figure C.2) and YBEY (Figure C.6) that both FR and LM identified cis associations for, TPP2 located on 13q32-q33 was found to be highly expressed in the lung tumor tissues. Golz et al. (2007) reports that the differential expression between diseased tissue lung tumor and healthy tissue lung demonstrates that the human TPP2 can be utilized to diagnose of respiratory diseases. Implementing FR, we identified a cis association at SNP rs660236 for TPP2 (Figure C.1). In Khodyrev et al. (2012), it is

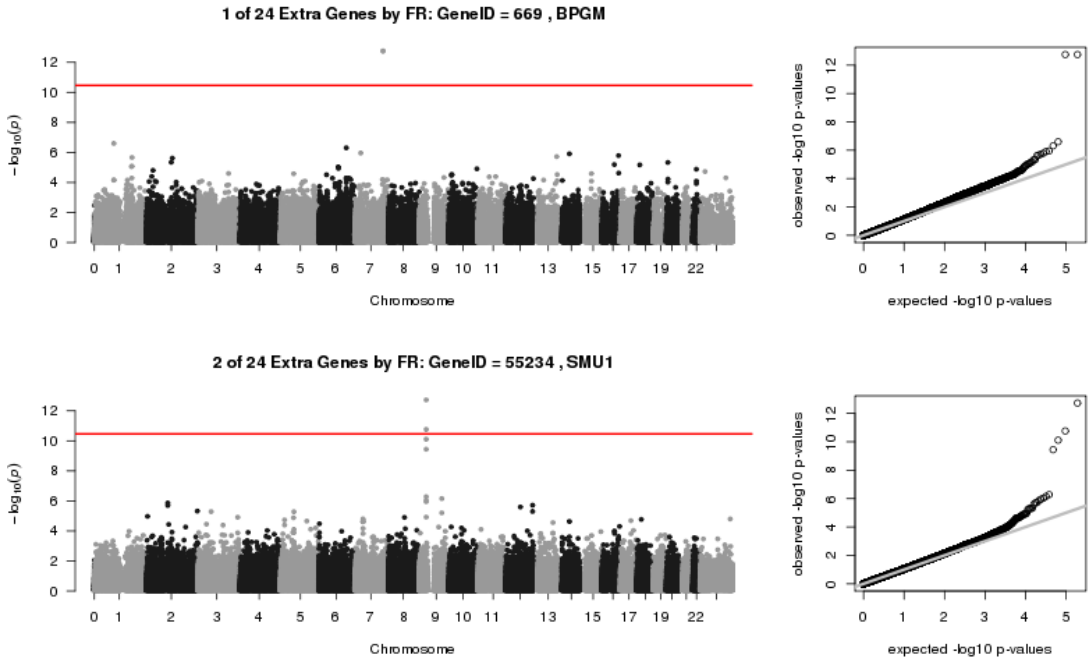


Figure 5.11: Manhattan and QQ plots for the top two genes for which a significant cis association was identified by FR but not by HEFT. A reference line at $-\log_{10}(0.05/(7,575 \times 191,959)) = 10.46$ for the Bonforroni threshold is indicated in the Manhattan plots.

concluded that NKIRAS1 gene on chromosome 3 can function as an oncogene in non-small cell lung cancer (NSCLC). FR detected a cis association at SNP rs11922577 for NKIRAS1 (Figure C.3). BTG3 that is located on 21q21.1 has been found to be significantly down-regulated in NSCLC (Chen et al., 2013). A cis association at SNP rs2824407 for BTG3 (Figure C.4) was identified by FR. A locus near MAN1C1 on chromosome 1 was detected to be associated with percent emphysemaamong a group of 2,510 African Americans in Manichaikul et al. (2014). We identified, via FR, a cis association at SNP rs12075373 for MAN1C1 (Figure C.5). Gene expression of CCDC15 on chromosome 11 was found to differentiate between aspirin-exacerbated respiratory disease (AERD) and aspirin-tolerant asthma (ATA) (Shin et al., 2012). We identified a cis association at SNP

rs11219878 for CCDC15 (Figure C.7). Analyses have pinpointed RGS17 as a major candidate for the familial lung cancer susceptibility locus on chromosome 6q23-25 region (You et al., 2009). FR revealed a cis association at SNP rs1933260 for RGS17 (Figure C.8). Also, differential expression of BPGM, a gene on chromosome 7q33, in planocellular lung cancer was demonstrated in Oparina et al. (2013). We identified, via FR, a cis association at SNP rs1160049 for BPGM (Figure 5.11). A summary of genes in the extra cis associations detected by our methods but not by HEFT that have previously been linked with lung disease is shown in Table 5.6.

We also investigated FR p-values for the four genes noted in Gao et al. (2014) as being of particular relevance to lung disease, namely GTF2H1, MTRR, TEFM, and RUVBL1, for each of which a cis eQTL was reported by HEFT but not by LM. In Figures 5.12 and 5.13, FR p-values and LM p-values for each of the four genes were compared. There are cis SNPs that were identified by FR as significantly associated with the genes whereas no strong signals were revealed by LM. Additionally, Manhattan and QQ plots produced with FR p-values are of the same pattern as those produced with HEFT p-values in Figure S15 of Gao et al. (2014). In fact, the list of top SNP associations for each of the four genes (Table 5.4) shows that FR found the same SNPs as HEFT with similar but more significant p-values than HEFT. Moreover, using the Bonferroni corrected threshold on the p-values, we see that FR found a significant eQTL association for genes GTF2H1 and MTRR while HEFT found a significant eQTL association for gene GTF2H1.

Implementing RRmix on the lung SAE data was feasible but it took 430 hours (i.e. 18 days) to complete in the same parallel R computing setting as FR. We

| HEFT | | | FR | | |
|---|---|---|---|---|---|
| Gene Symbol | SNP | p-value | Gene Symbol | SNP | p-value |
| GTF2H1 | rs4150622 | 2.148023e-12 | GTF2H1 | rs4150622 | 9.903150e-13 |
| MTRR | rs179171 | 2.819637e-10 | MTRR | rs179171 | 2.737549e-11 |
| TEFM | rs9889887 | 1.090420e-09 | TEFM | rs9889887 | 2.019375e-10 |
| TEFM | rs9898084 | 1.090420e-09 | TEFM | rs9898084 | 2.019375e-10 |
| TEFM | rs9895785 | 1.090420e-09 | TEFM | rs9895785 | 2.019375e-10 |
| TEFM | rs7214570 | 1.090420e-09 | TEFM | rs7214570 | 2.019375e-10 |
| RUVBL1 | rs12695504 | 1.225830e-09 | RUVBL1 | rs12695504 | 1.015016e-09 |

Table 5.4: Top SNP associations for genes GTF2H1, MTRR, TEFM, and RUVBL1 identified by HEFT and FR (right).

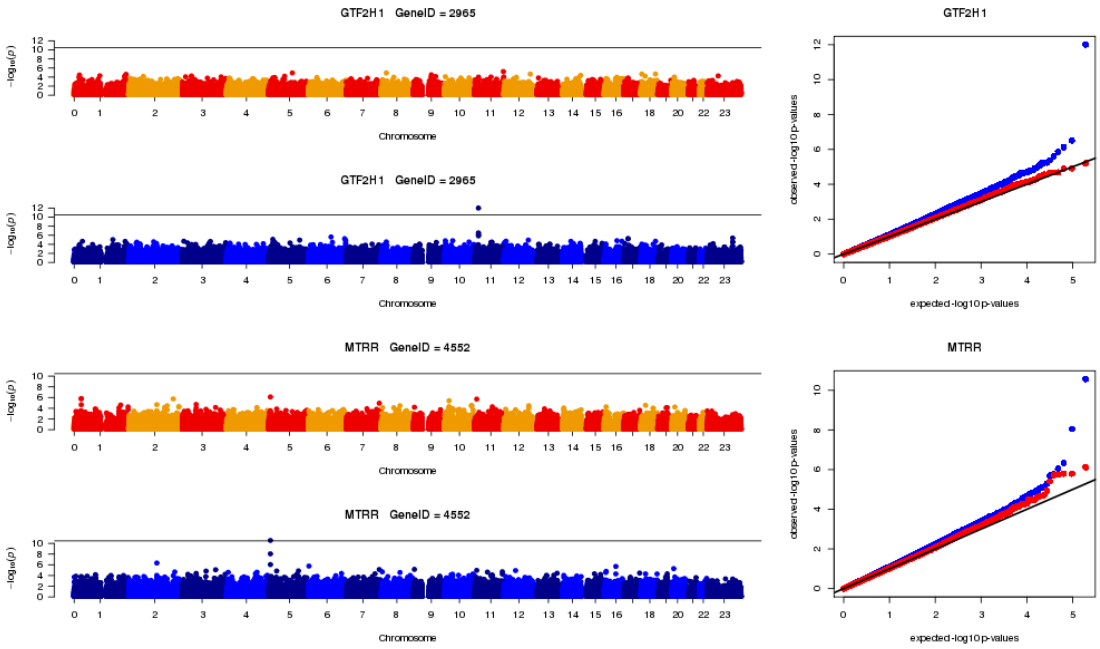


Figure 5.12: Manhattan and QQ plots on p-values computed via LM (red/orange) and FR (blue) for genes GTF2H1 and MTRR. A reference line at $-\log_{10}(0.05/(7,575 \times 191,959)) = 10.46$ for the Bonforroni threshold is indicated in the Manhattan plots.

note that results of RRmix and FR are not directly comparable due to the fact that inference for RRmix is based on the posterior mean of the gene-specific latent indicator rather than p-values. Nevertheless, using a threshold of 0.9999, we obtained 2,152 significant associations found by RRmix, among which 306

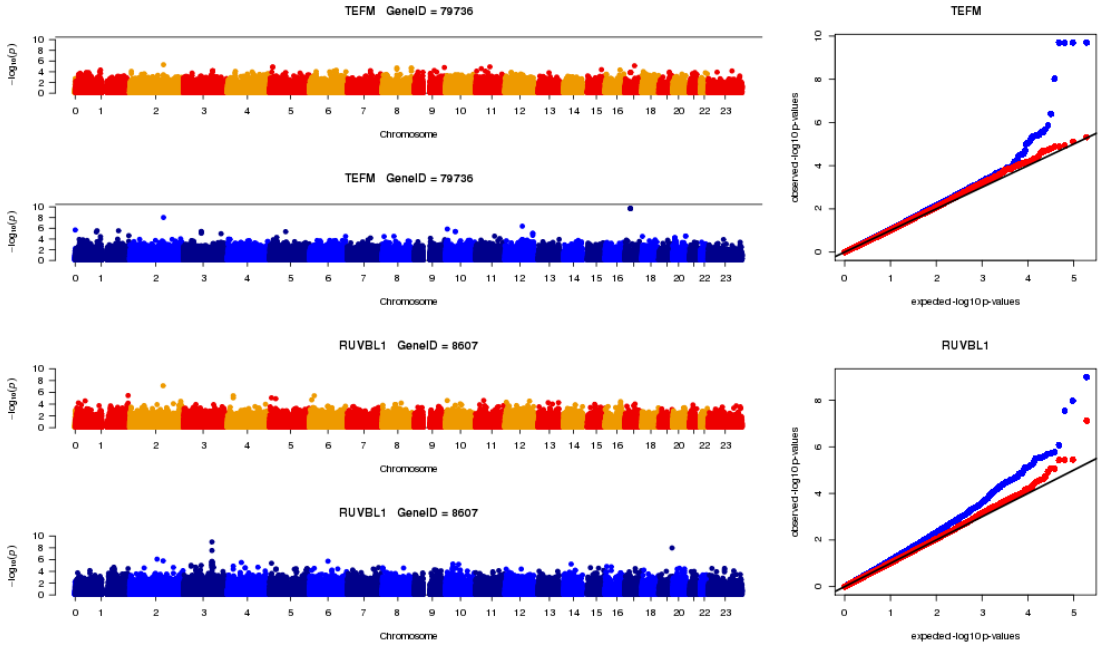


Figure 5.13: Manhattan and QQ plots on p-values computed via LM (red/orange) and FR (blue) for genes TEFM and RUVBL1. A reference line at $-\log_{10}(0.05/(7,575 \times 191,959)) = 10.46$ for the Bonforroni threshold is indicated in the Manhattan plots.

are cis, 358 are non-duplicated, and 107 are cis and non-duplicated. There are 71 common associations in the top 100 non-duplicated eQTL associations found by RRmix (C.2) and the top 100 non-duplicated eQTL associations found by FR (C.1). The 2,152 significant associations RRmix detected contain all 96 of the non-duplicated gene-SNP associations discovered by HEFT, as well as all but 3 of the 217 cis associations discovered by FR - in fact, those 3 cis associations are at 3 different genes, and except for 1 gene that RRmix completely missed, RRmix detected 1 less SNP than FR for each of the two remaining genes. Among the 61 cis and non-duplicated associations found by HEFT, RRmix recovered all 61 cis associations, 46 of which at the exact same SNP as detected by HEFT. Additionally, there are 46 cis and non-duplicated associations discovered by RRmix but

not by HEFT, among which 22 were discovered by FR but not by HEFT. The one gene that RRmix completely missed when comparing cis associations with FR is UBA1, which is one of the 2 cis and non-duplicated associations detected by FR but not by RRmix or HEFT, and that cis association detected by FR appears to be a biological false positive (Figure C.1).

Information on the GeneCards encyclopedia (www.genecards.org) indicates that several genes in cis associations identified by RRmix are related to lung disease. CHRNA10 on chromosome 11 is known to be associated with nicotine dependence, a disease related to lung cancer. It is noted in Saccone et al. (2010) that SNPs near CHRNA10 show modest association with nicotine dependence risk in a sample of 710 African Americans. RRmix identifies a cis association at SNP rs2672216 for CHRNA10. CLSTN2 on chromosome 3q23 is known to be associated with pulmonary function. Variation in CLSTN2 was reported to be associated with smokers ability to achieve and sustain abstinence from smoking (Uhl et al., 2008). RRmix identifies a cis association at SNP rs6766541 for CLSTN2. Additionally, RRmix identifies a cis association for each of the following genes known to be associated with pneumonia and/or tuberculosis: DHRS7 on chromosome 14q23.1, DIP2C on chromosome 10p15.3, and NARS2 on chromosome 11p14.1. Additionally, RRmix identifies a cis association for NUP210 on chromosome 3p25.1 which is known to be associated with influenza.

That RRmix did not detect any significant association for gene UBA1 for which the cis association detected by FR is a likely false positive, that RRmix reports more cis and non-duplicated associations than FR and HEFT, and such associations include genes that are of relevance to lung disease suggest that RRmix has the potential to achieve superior accuracy with a sensible threshold for pos-

| Gene ID | Gene Symbol | SNP | Chr (SNP) | Position | Method | Disease | Reference on Lung Disease Association |
|---|---|---|---|---|---|---|---|
| 669 | BPGM | rs1160049 | 7 | 133960652 | FR, RRmix | planocellular lung cancer | Oparina et al. (2013) |
| 7174 | TPP2 | rs660236 | 13 | 102123999 | FR, RRmix | lung tumor | Golz et al. (2007) |
| 11005 | SPINK5 | rs4529181 | 5 | 147449164 | FR, RRmix, LM | asthma in adults | Accordini et al. (2013) |
| 28512 | NKIRAS1 | rs11922577 | 3 | 23965245 | FR | non-small cell lung cancer | Khodyrev et al. (2012) |
| 28512 | NKIRAS1 | rs4490300 | 3 | 23950585 | RRmix | non-small cell lung cancer | Khodyrev et al. (2012) |
| 10950 | BTG3 | rs2824407 | 21 | 17930971 | FR, RRmix | non-small cell lung cancer | Chen et al. (2013) |
| 57134 | MAN1C1 | rs12075373 | 1 | 25808837 | FR, RRmix | emphysema | Manichaikul et al. (2014) |
| 54059 | YBEY | rs4818835 | 21 | 46549130 | FR, RRmix, LM | pneumonia and tuberculosis | the GeneCards encyclopedia (www.genecards.org) |
| 80071 | CCDC15 | rs11219878 | 11 | 124427978 | FR, RRmix | aspirin-exacerbated respiratory disease and aspirin-tolerant asthma | Shin et al. (2012) |
| 26575 | RGS17 | rs1933260 | 6 | 153421070 | FR, RRmix | familial lung cancer | You et al. (2009) |
| 57053 | CHRNA10 | rs2672216 | 11 | 3648600 | RRmix | nicotine dependence | Saccone et al. (2010) |
| 64084 | CLSTN2 | rs6766541 | 3 | 141549924 | RRmix | pulmonary function | Uhl et al. (2008) |
| 51635 | DHRS7 | rs425565 | 14 | 59673636 | RRmix | pneumonia and tuberculosis | the GeneCards encyclopedia (www.genecards.org) |
| 22982 | DIP2C | rs816628 | 10 | 609170 | RRmix | tuberculosis | the GeneCards encyclopedia (www.genecards.org) |
| 79731 | NARS2 | rs4945303 | 11 | 77961451 | RRmix | pneumonia | the GeneCards encyclopedia (www.genecards.org) |
| 23225 | NUP210 | rs9817742 | 3 | 13431803 | RRmix | influenza | the GeneCards encyclopedia (www.genecards.org) |

Table 5.6: Cis and non-duplicated associations detected by our methods and LM but not by HEFT that involve genes that have previously been linked with lung disease. As cis and non-duplicated associations were considered, FR and RRmix identified different but nearby SNPs for gene NKIRAS1.

terior mean of latent indicators despite its less attractive computational speed than FR.

### Verification of adjustment for hidden effects

For each of the three methods LM, HEFT, and FR, a heat map of p-values for the first 10,000 SNPs and all 7,575 genes was produced. The data set containing p-values of HEFT was downloaded from the site at NCBI's Gene Expression

Omnibus where authors of Gao et al. (2014) deposited their results. As seen in the three heat maps in Figure 5.14, FR and HEFT are capable of removing, to matchable extent, the global impact of hidden factors that are non-orthogonal to genotypes. Such global impact is present in the heat map of LM and can be visualized as several vertical lines each of which indicates strong associations between a SNP and the majority of the genes.

Additionally, we conducted an experiment similar to the recovery of smoking effect experiment in supplementary materials of Gao et al. (2014) to verify that our FR and RRmix models are capable of adjusting for hidden effects in the lung SAE data. That is, FR and RRmix models are each fitted to the data with the smoking status variable removed from the known covariates, as if smoking status is among the hidden factors that are to be learned by the models. If the effect of smoking is "recovered" in the plot of pairs of the first few factor loadings estimated by the models, this is clear indication that the models adjust for the smoking effect correctly. Indeed, clear separation according to smoking status is visible in the plots of the first three factor loadings estimated by FR and by RRmix (Figure 5.15) when the models were implemented on the lung SAE data: all 7,575 genes and SNP rs1886498, the first SNP in the 96 non-duplicated associations reported in Gao et al. (2014).

### 5.2.2 Differential expression analysis of Microarray data with Alzheimer's Disease Data

Gagnon-Bartsch and Speed (2012) presented gender study results on the Alzheimer's disease data that was analyzed in a microarray study in Blalock

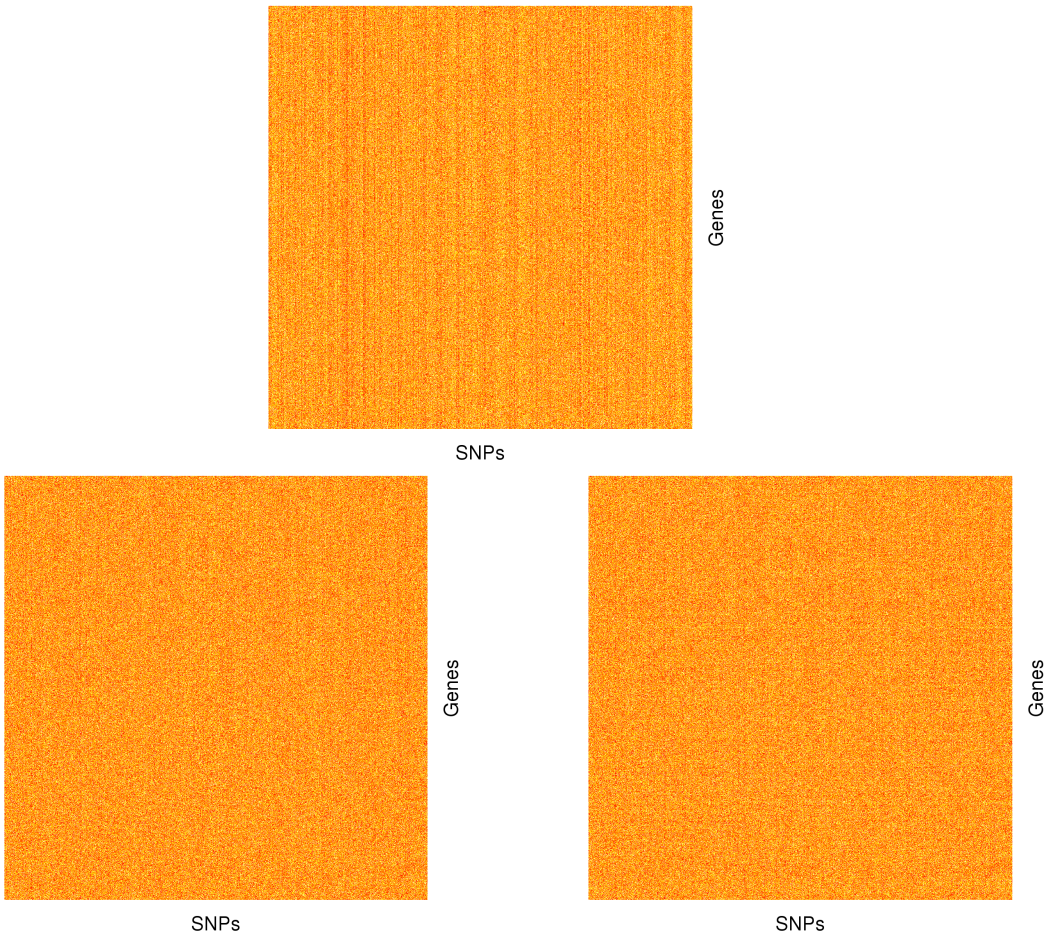


Figure 5.14: A heat map of the subset of p-values that correspond to the first 10,000 SNPs (x-axis) and all 7,575 genes (y-axis) computed by LM (top), HEFT (bottom-left), and FR (bottom-right). The darker red a pixel, the smaller the p-value. Vertical lines are visible in the heat map for LM but not in the heat map for HEFT or FR, indicating successful removal of global impact of hidden factors while identifying significant associations by HEFT and by FR.

et al. (2004). The RUV-2 method of Gagnon-Bartsch and Speed (2012) involves an analogous model to HEFT in which estimation of hidden factors is implemented via factor analysis, but RUV-2 relies on so-called negative control genes, known to be null genes with respect to the biological factor of interest, in order to estimate hidden factors, and in turn adopts a two-step parameter estimation

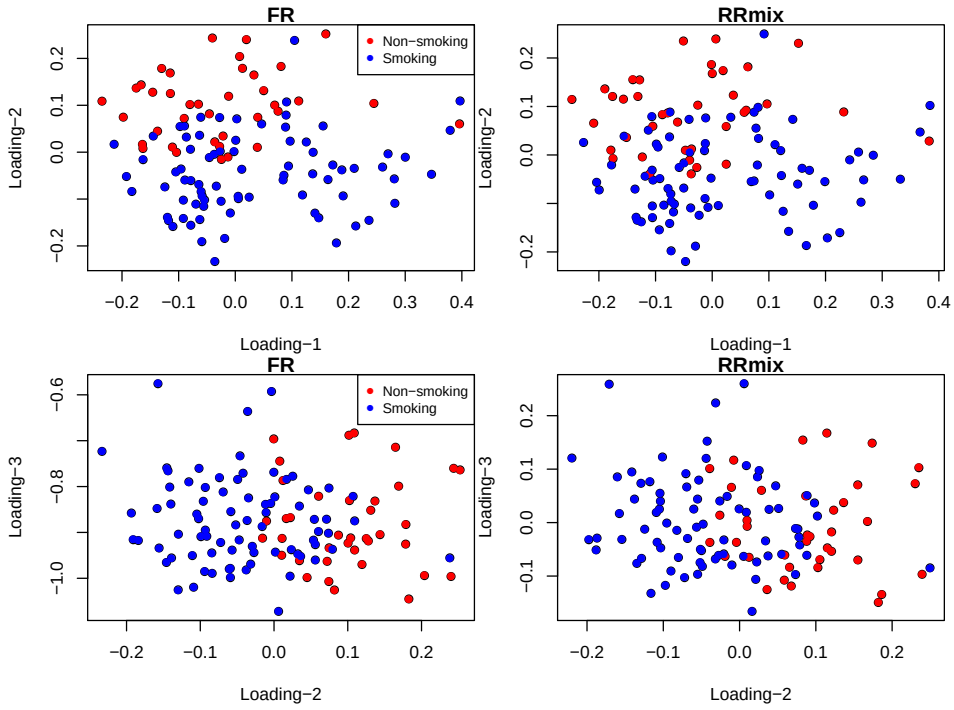


Figure 5.15: Plot of loading 1 and loading 2 (top), and of loading 2 and loading 3 (bottom), estimated by FR and by RRmix, when the models were implemented on the lung SAE data: all 7,575 genes and SNP rs1886498, the first SNP in the 96 non-duplicated associations reported in Gao et al. (2014). Separation of smokers (blue) and non-smokers (red) is clear in all plots, indicating correct recovery of smoking effect by FR and by RRmix when smoking status was treated as hidden.

procedure. RUV-2 discovers genes that are differentially expressed in the brain with respect to gender, and its performance on the Alzheimer's disease data, as shown in Section A of the supplementary material of Gagnon-Bartsch and Speed (2012), demonstrates that RUV-2 is capable of removing unwanted variation with the aid of negative control genes and providing adequate adjustment to gene expression data in differential expression analysis. Note that the true non-null genes in this case are known in some sense, since genes that come from the X or Y chromosome are expected to be differentially expressed between men

and women.

We obtained the publicly available Alzheimer's disease data on GEO (GSE1297). Microarray data on 22,283 genes for each of the 31 individuals who suffered from various stages of Alzheimer's disease are available for analysis. Originally, there were 35 individuals, but Blalock et al. (2004) discarded 4 samples due to unclear disease severity. Unlike the authors of Gagnon-Bartsch and Speed (2012), we were able to directly obtain gender information of each individual on the GEO web site. However, in the data we obtained there are 18 women and 13 men, and this contradicts the 19 women and 12 men that Gagnon-Bartsch and Speed (2012) found via expression levels of XIST and DDX3Y genes, as well as the 16 women and 19 men out of the original 35 individuals considered in Blalock et al. (2004).

We show the number of X / Y genes, i.e. genes that come from the X or Y chromosome, out of the top 20, 40, 60 non-null genes detected by RRmix, FR, HEFT-R, RUV-2, and LM in Table 5.7. Genes are ranked in the order of increasing p-value for FR, HEFT-R, RUV-2, and LM, and in the order of decreasing posterior mean of the latent indicator for RRmix.

With the gene expression data being the response variable, a SNP vector in eQTL analysis acts in the same way as the categorical variable of interest in differential expression analysis, hence our models RRmix and FR are applicable for the gender study on the Alzheimer's disease data, where gender was included in the model as the one and only "SNP". There was no known covariate in this gender study. The number of hidden factors for RRmix and FR was chosen to be 6, which was determined based on visual examination of the scree plot (Figure 5.16) of principal component analysis on the gene expression correlation matrix.

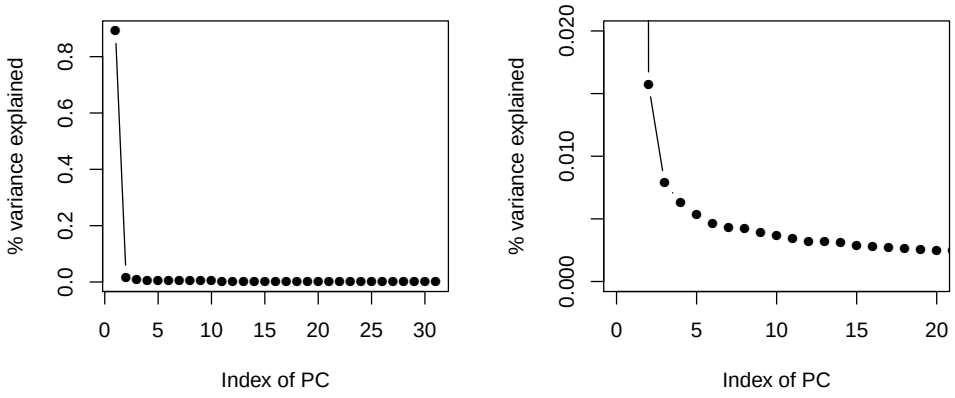


Figure 5.16: Scree plot of all PCs (left) and the second to the twentieth PCs (right) based on principal component analysis on the gene expression correlation matrix of the Alzheimer's disease data.

The publicly available HEFT application was written in C++ by authors of Gao et al. (2014) and it does not cater for scenarios where no known covariates are included in the model. Therefore we adapted the HEFT-R implementation to a replicate of HEFT that allows for absence of known covariates.

RUV-2 was implemented with the R code provided in Section G of the supplementary material of Gagnon-Bartsch and Speed (2012), in which SVD for estimation of hidden factors and Limma (Smyth, 2004) for calculation of p-values were employed. Using the list of housekeeping genes in Eisenberg and Levanon (2003), we found 1,107 housekeeping genes among the 22,283 genes and used them as negative controls, which was different to the 1,112 housekeeping genes used as negative controls by authors of Gagnon-Bartsch and Speed (2012) who also referred to Eisenberg and Levanon (2003). Despite of the discrepancy in the number of housekeeping genes and in the gender variable noted above, we used 10 hidden factors for RUV-2, same as in Gagnon-Bartsch and Speed (2012), in the hope that we recover results of RUV-2 produced by the authors to the

fullest extent possible which can then serve as reference.

All five methods were implemented on gene expression data that did not go through any preprocessing noted in Gagnon-Bartsch and Speed (2012) and were simply $\log_2$ transformed.

| | Top 20 | Top 40 | Top 60 |
|---|---|---|---|
| RRmix | **14** | 14 | 16 |
| FR | **14** | 14 | 14 |
| HEFT-R | 13 | 14 | 14 |
| LM | 13 | 13 | 13 |
| RUV-2 (k=10) | 12 | **16** | **18** |

Table 5.7: Number of X / Y genes found in the top 20 / 40 / 60 by RRmix, FR, HEFT-R, LM, and RUV-2 in the Alzheimer's study. All procedures were given unpreprocessed data as input. RRmix, FR, HEFT-R were implemented with 6 hidden factors and RUV-2 with 10 hidden factors. The highest number in each column is shown in bold.

RRmix, as seen in Table 5.7, detected the closest numbers of top X / Y genes to RUV-2 among the four methods. Differences of a few genes are non-trivial especially in the context of differential expression analysis where top genes are identified for use in further study. Nonetheless, RRmix achieved performance close to RUV-2 without resorting to control gene information. Also without the use of control genes, FR and HEFT-R had matching performance. All of RRmix, FR, and HEFT-R detected more top X / Y genes than LM, which verifies the ability of accounting for unwanted variation of these methods.

### 5.2.3 Count-based differential expression analysis of RNA-sequencing data with Zebrafish Data

The zebrafish data in Risso et al. (2014) is a set of RNA-sequencing (RNA-seq) count data that was used to evaluate performance of the authors' RUV family of methods in removing unwanted variation in RNA-seq data in the context of differential expression analysis. There are 20,865 genes and 6 samples, 3 control replicate samples and 3 treatment replicate samples in the gene-level count data. Details of data collection and filtering are in Risso et al. (2014). We gained access to the zebrafish data as well as the source code of the RUV methods via the Bioconductor R packages noted in Supplementary Software of Risso et al. (2014). We implemented RUVg, a method of the RUV class that models the count data using a negative-binomial distribution and makes use of control genes to estimate hidden factors, and hence can be viewed as the discrete version of RUV-2 (Gagnon-Bartsch and Speed, 2012). Adopting the same settings for RUVg as in Risso et al. (2014), i.e. 15,839 empirical control genes, 1 hidden factor, and no upper-quartile (UQ) normalization, we were able to reproduce the claimed 475 differentially expressed genes detected by RUVg.

RRmix and FR, as well as HEFT-R, can be applied in this case for differential expression analysis on the RNA-seq count data. As these models were built for continuous response variables, we transformed each count by first adding 1 and then taking $\log_2$ of its proportion in the replicate sample. The variable of interest was the binary treatment variable. No covariates were involved in fitting the models. The number of hidden factors was chosen to be 2, based on visual examination of the scree plot, similar to Figure 5.16, of principal component analysis on the correlation matrix of the gene-level log-counts. Although

it is likely that the differentially expressed genes found by RUVg contain false positives, without ground truth information we treated the 475 differentially expressed genes found by RUVg as a gold standard, and evaluated classification performance of RRmix and FR.

The top 475 genes detected to be differentially expressed with respect to treatment by RRmix, FR, and HEFT-R have, respectively, 208, 148, and 77 genes in common with the top 475 genes of RUVg. ROC curves (Figure 5.17) with RUVg results used as "truth" show that RRmix achieves the highest power for small values of false positive rate, and FR is associated with the largest value of area under ROC curve (AUC) at 0.858 while AUC for RRmix is 0.778 and for HEFT-R is 0.761. Additionally, no clustering of the samples according to the treatment effect is visible in the plot of the two hidden factor loadings estimated by RRmix, FR, and HEFT-R (Figure 5.18). This verifies that no variation attributed to the treatment effect was incorrectly captured by the estimated hidden factors. Furthermore, the estimated hidden factor loadings in (Figure 5.18) explain the captured unwanted variation, whereas the first two PCs in plot (c) of Figure 2 in Risso et al. (2014) explain total variation. Plot (c) of Figure 2 in Risso et al. (2014) indicates that there exists unwanted variation which prevents clustering by treatment. The resemblance between the two sets of plots shows that our methods capture the same unwanted variation as evident in plot (c) of Figure 2 in Risso et al. (2014).

Implementation on the zebrafish data illustrated that our approaches can be adapted for classification in which count data are involved. Given that estimation of hidden effects in RRmix and FR does not rely on any negative control genes, our methods perform well in classification of count data while simulta-

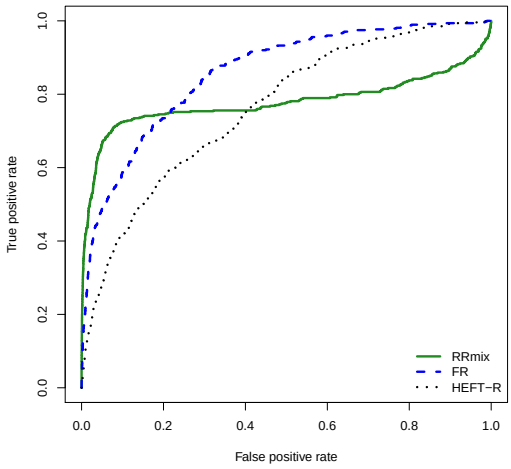


Figure 5.17: ROC curves for RRmix (solid), FR (dashed), and HEFT-R (dotted) with the 475 differentially expressed genes detected by RUVg as a gold standard.

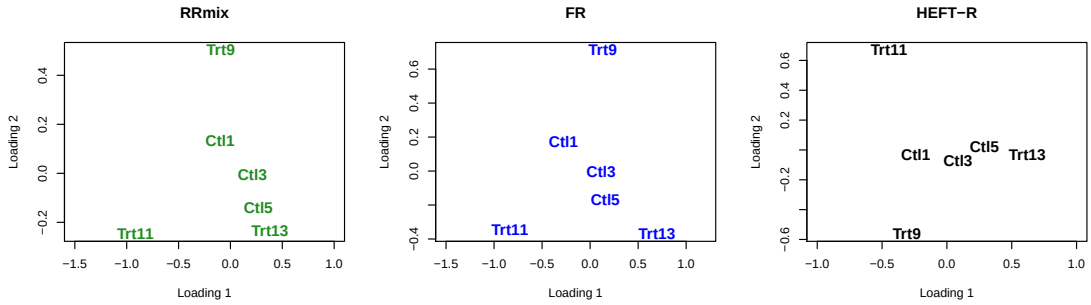


Figure 5.18: Plot of the two hidden factor loadings estimated by RRmix, FR, and HEFT-R (left to right). No clustering by treatment is observed, which indicates little "wanted" variation was incorrectly captured in the estimated hidden factors by the methods, and hence is desirable.

neously accounting for hidden effects.

CHAPTER 6

# DISCUSSION

Classification problems in bioinformatics applications are often solved via hierarchical model-based methods. One of the challenges lies in the high-dimensionality of data which can be a computational barrier to the feasibility and efficiency of methods. For example, fully Bayesian inference via MCMC is often computationally infeasible. An alternative computational approach for fully Bayesian inference is variational Bayes (VB). In this dissertation, it is demonstrated that VB can reliably approximate marginal posterior distributions required for classification problems arising in bioinformatics.

VB has been promoted in the statistical and computer science literature as an alternative to MCMC for Bayesian computation. However, the performance and feasibility of VB has not been widely investigated for hierarchical mixture models used in sparse classification problems. Part of this dissertation demonstrates the implementation of VB in that context and shows that it is capable of efficiently producing reliable solutions with classification performance comparable to much slower MCMC methods. A known issue with VB is its tendency to underestimate variability in marginal posteriors. Grid-based variational approximation (GBVA) (Ormerod, 2011) is designed to correct this, but its performance in the context of hierarchical mixture models is not promising either in terms of accuracy or computational speed.

As the demand for high-dimensional data analysis tools grow, the search for fast and accurate alternatives to MCMC continues to be an important open research area. Other deterministic alternatives to MCMC in posterior density approximation include expectation propagation (EP) (Minka, 2001a,b) and inte-

grated nested Laplace approximation (INLA) (Rue et al., 2009). In EP, while a similar factorization of the joint posterior density to that in VB is assumed, the approximate posterior is found by moment matching implied by minimization of the reverse form of Kullback-Leibler divergence to that used in VB theory. However, as noted in Bishop (2006), drawbacks of EP include lack of a guarantee of convergence and failure to capture any mode if the true posterior density is multimodal. For the class of latent Gaussian models, approximate Bayesian inference can be achieved via INLA (Rue et al., 2009). The INLA method relies on Gaussian approximations and numerical methods, and does not require an assumption concerning factorization of the joint density. However, a limitation is the fact that models that contain non-Gaussian latent variables in the linear predictor do not belong to the class of latent Gaussian models for which INLA is applicable.

VB has been criticized (Rue et al., 2009) for its applicability being restricted to the class of conjugate-exponential models (Beal, 2003). However, provided that VB is feasible for a reparameterized model whose inference is close to the original model (Beal, 2003), VB is often a good approach for fully Bayesian inference because of its relative ease of implementation and computational speed.

Another challenge that comes with high-dimensionality of data in classification problems in bioinformatics is the presence of unwanted variation in the data. Without effective methods that account for unwanted variation, model-based classification methods generally suffer from low power and high false discovery rate. HEFT (Gao et al., 2014) appears to be an efficient and relatively accurate tool that was developed in the context of eQTL studies. Parts of the implementation of HEFT appear to rely on ad hoc assumptions. We developed

the hierarchical mixture model RRmix that has been shown to perform well as a classifier while simultaneously accounting for hidden effects in cases where its implementation is feasible. We evaluated a family of models as alternatives to RRmix that are more computationally efficient for real data applications. Among these the FR model is shown to be a statistically reliable model which can be implemented efficiently using EM algorithm. Compared with HEFT, FR involves less restrictive and more statistically transparent model assumptions.

Other Bayesian model-based classification methods designed in the context of eQTL mapping include Stegle et al. (2010) and Scott-Boyer et al. (2012) and those reviewed therein. Models in Stegle et al. (2010) and Scott-Boyer et al. (2012) are fully Bayesian hierarchical mixture models where the complete gene expression data and SNP data are analyzed at once with a latent indicator of gene-SNP association specific to each gene-SNP pair. Approximate posterior inference is conducted via variational Bayes in the variational Bayesian QTL mapper (VBQTL) method of Stegle et al. (2010) which, similarly to RRmix, learns SNP effect while accounting for hidden effects simultaneously. Posterior exploration of the integrated hierarchical Bayesian model (iBMQ) in Scott-Boyer et al. (2012) is performed via MCMC. However, iBMQ does not account for hidden effects.

A known concern about methods that use some form of factor analysis to model unwanted variation is referred to as "throwing out the baby with the bathwater" by authors of Gagnon-Bartsch and Speed (2012). That is, it is possible that the captured unwanted variation contains a part of wanted variation that is due to the biological variable of interest, which in turn causes degradation in accuracy. The RUV class of methods in Gagnon-Bartsch and Speed

(2012); Jacob et al. (2012); Gagnon-Bartsch et al. (2013); Risso et al. (2014) attempt to avoid such a problem by using negative controls to target and isolate sources of unwanted variation. Negative control genes are genes that are known to be unassociated with the biological variable of interest. Negative control samples are typically replicate samples whose differences are not affected by the variable of interest. While these methods may serve as reasonable data normalization methods the purpose of which is to remove unwanted variation from the data, they do not account for estimation error of the unwanted variation when treating the inferred unwanted variation as known for subsequent analysis. For example, for differential expression analysis, after the inferred unwanted variation is plugged in to the model in step two of the RUV-2 approach, there is a risk that bias in estimation of unwanted variation is introduced but is not addressed in detection of differentially expressed genes. Moreover, in practice, the quality of negative controls, i.e. the degree to which the effect is truly zero, may require further assessment, especially in the event that negative controls may not be readily available and empirical negative controls are used instead.

By simultaneously learning hidden effects with effect of the biological variable of interest instead of adopting a two-step strategy, our family of models do not rely on negative controls to direct estimation of unwanted variation but rather let the entire gene expression data speak for itself via an empirical Bayes approach accomplished by the EM algorithm. Instead of making assumptions on orthogonality between the variable of interest and the hidden factors or on how much more or less variation is explained by the hidden factors than the variable of interest, we allow our estimation procedures to seek $q$ many hidden factors from the full data while simultaneously estimating the effect of the primary variable and error variance with the aid of shrinkage estimation. This

would be particularly desirable when little is known about true underlying hidden effects and it is possible that the set of true hidden factors are in fact correlated with the variable of interest in a complex way. The quality of estimated hidden effects may be assessed in several ways, two of which are 1) the estimated hidden effects should capture true hidden effects correctly, and 2) the estimated hidden effects should not include or mask the primary effects. Experiments such as recovery of the smoking effect in Section 5.2.1 are helpful in demonstrating the ability to capture hidden effects, and observing little primary effect among estimated hidden effects as demonstrated in Section 5.2.3 assures that "throwing out the baby with the bathwater" has been avoided. That said, mathematical quality assessment of estimated hidden effects remain to be investigated.

Other future work include choosing number of hidden factors based on statistical criteria and automation of this process, and further development of software for RRmix and FR.

APPENDIX A

**TECHNICAL DETAILS OF CHAPTER 3**

## A.1 VB-LEMMA algorithm

### A.1.1 The B-LEMMA model

We consider a natural extension to the LEMMA model in Bar et al. (2010): a fully Bayesian three-component mixture model, B-LEMMA:

$$
\begin{aligned}
d_g|(b_{1g}, b_{2g}), \psi_g, \sigma^2_{\epsilon,g}, \tau &= \tau + (b_{1g} - b_{2g})\psi_g + \epsilon_g \\
m_g|\sigma^2_{\epsilon,g} &\sim \frac{\sigma^2_{\epsilon,g}\chi^2_{f_g}}{f_g}, \quad \text{where } f_g := n_{1g} + n_{2g} - 2 \\
\psi_g|\psi, \sigma^2_\psi &\sim N(\psi, \sigma^2_\psi) \quad i.i.d. \\
\epsilon_g|\sigma^2_{\epsilon,g} &\sim N(0, \sigma^2_g), \quad \text{where } \sigma^2_g := \sigma^2_{\epsilon,g} c_g, \quad c_g := \frac{1}{n_{1g}} + \frac{1}{n_{2g}} \\
(b_{1g}, b_{2g}, 1 - b_{1g} - b_{2g})|(p_1, p_2) &\sim \textit{Multinomial}\,(1; p_1, p_2, 1 - p_1 - p_2) \quad i.i.d. \\
\tau &\sim N\left(\mu_{\tau_0}, \sigma^2_{\tau_0}\right) \\
\psi &\sim N\left(\mu_{\psi_0}, \sigma^2_{\psi_0}\right) \\
\sigma^2_\psi &\sim IG\left(A_\psi, B_\psi\right) \\
\sigma^2_{\epsilon,g} &\sim IG\,(A_\epsilon, B_\epsilon) \quad i.i.d. \\
(p_1, p_2, 1 - p_1 - p_2) &\sim \textit{Dirichlet}\,(\alpha_1, \alpha_2, \alpha_0)\,,
\end{aligned}
$$

where $(b_{1g}, b_{2g})$ takes values $(1, 0), (0, 1)$, or $(0, 0)$, indicating that gene $g$ is in non-null group 1, non-null group 2, or null group, respectively. $p_1$ and $p_2$ are proportions of non-null group 1 and non-null group 2 genes. Hence, the non-null proportion is $p_1 + p_2$. Each of $\tau$ and $\psi_g$ represents the same quantity as in the B-LIMMA model.

### A.1.2 Algorithm

The VB-LEMMA algorithm was derived based on an equivalent model to B-LEMMA. In the equivalent model, the gene-specific treatment effect is treated as the combination of a fixed global effect and a random zero-mean effect. That is, conditional distribution of $d_g$ and that of $\psi_g$ are replaced with

$$d_g|(b_{1g}, b_{2g}), u_g, \sigma^2_{\epsilon,g}, \tau = \tau + (b_{1g} - b_{2g})\psi + (b_{1g} + b_{2g})u_g + \epsilon_g$$

$$u_g|\sigma^2_\psi \sim N(0, \sigma^2_\psi) \quad i.i.d..$$

The set of observed data and the set of unobserved data are identified as

$$\mathbf{y} = \{\{d_g\}, \{m_g\}\}$$

$$H = \{\{\boldsymbol{b}_g\}, \{u_g\}, \{\sigma^2_g\}, \tau, \psi, \sigma^2_\psi, \boldsymbol{p}\}$$

where $\boldsymbol{b}_g = (b_{1g}, b_{2g})$ and $\boldsymbol{p} = (p_1, p_2)$.

Because of the similarities of the B-LEMMA model to the B-LIMMA model, derivation of VB-LEMMA was achieved by extending the derivation of VB-LIMMA that involves a gene-specific zero-mean random effect parameter. The VB algorithm based on the exact B-LEMMA model was also derived for comparison. However, little discrepancy in performance between the VB algorithm based on the exact model and VB-LEMMA was observed. Therefore, VB-LEMMA based on the equivalent model was adopted.

The product density restriction

$$q(H) = q_{\{\boldsymbol{b}_g\}}(\{\boldsymbol{b}_g\}) \times q_{\{\psi_g\}}(\{\psi_g\}) \times q_{\{\sigma^2_g\}}(\{\sigma^2_g\}) \times q_{(\tau,\boldsymbol{p})}(\tau, \boldsymbol{p}) \times q_{(\psi,\sigma^2_\psi)}(\psi, \sigma^2_\psi)$$

leads to $q$-densities

$$q_\tau(\tau) = N\left(\widehat{M_\tau}, \widehat{V_\tau}\right)$$

$$q_\psi(\psi) = N\left(\widehat{M_\psi}, \widehat{V_\psi}\right)$$

$$q_{\sigma^2_g}(\sigma^2_g) = IG\left(A_{\sigma^2_g}, \widehat{B_{\sigma^2_g}}\right)$$

$$q_{(b_{1g},b_{2g},1-b_{1g}-b_{2g})}\left(\left(b_{1g},b_{2g},1-b_{1g}-b_{2g}\right)\right) = Multinomial\left(\widehat{M_{b_{1g}}},\widehat{M_{b_{2g}}},1-\widehat{M_{b_{1g}}}-\widehat{M_{b_{2g}}}\right)$$

$$q_{(p_1,p_2,1-p_1-p_2)}((p_1,p_2,1-p_1-p_2)) = Dirichlet\left(\widehat{\alpha_{p_1}},\widehat{\alpha_{p_2}},\widehat{\alpha_{p_0}}\right)$$

$$q_{\sigma_\psi^2}(\sigma_\psi^2) = IG\left(A_{\sigma_\psi^2},\widehat{B_{\sigma_\psi^2}}\right).$$

It is only necessary to update the variational posterior means $\hat{M}_{\cdot}$ in VB-LEMMA. Upon convergence, the other variational parameters are computed based on the converged value of those involved in the iterations. The iterative scheme is as follows:

i) Initialize

$$\widehat{M_{\sigma_\psi^{-2}}} = 1, \qquad \widehat{M_{\sigma_g^{-2}}} = \frac{1}{c_g}\ \forall\ g, \qquad \widehat{M_{u_g}} = 0\ \forall\ g$$

$$\widehat{M_{b_g}} = \begin{cases} (1,0,0) & \text{if rank}(d_g) \geq (1-0.05)G \\ (0,1,0) & \text{if rank}(d_g) \leq 0.05G \\ (0,0,1) & \text{otherwise} \end{cases} \qquad \text{for each } g$$

$$\widehat{M_\psi} = \frac{1}{2}\left(\left|\sum_{\{g:\text{rank}(d_g)\geq(1-0.05)G\}} d_g - \sum_{g=1}^{G} d_g\right| + \left|\sum_{g=1}^{G} d_g - \sum_{\{g:\text{rank}(d_g)\leq 0.05G\}} d_g\right|\right)$$

Set $A_{\sigma_\psi^2} = \dfrac{G}{2} + A_\psi$ and $A_{\sigma_g^2} = \dfrac{1+f_g}{2} + A_\varepsilon$ for each $g$.

ii) Update

$$\widehat{M_\tau} \leftarrow \left\{\sum_g \widehat{M_{\sigma_g^{-2}}}\left[\left(1-\widehat{M_{b_{1g}}}-\widehat{M_{b_{2g}}}\right)d_g + \widehat{M_{b_{1g}}}\left(d_g-\widehat{M_\psi}-\widehat{M_{u_g}}\right) + \widehat{M_{b_{2g}}}\left(d_g+\widehat{M_\psi}-\widehat{M_{u_g}}\right)\right] + \frac{\mu_{\tau_0}}{\sigma_{\tau_0}^2}\right\} \times \frac{1}{\sum_g \widehat{M_{\sigma_g^{-2}}} + \frac{1}{\sigma_{\tau_0}^2}}$$

$$\widehat{M_\psi} \leftarrow \left\{\sum_g \widehat{M_{\sigma_g^{-2}}}\left(\widehat{M_{b_{1g}}}-\widehat{M_{b_{2g}}}\right)\left(d_g-\widehat{M_\tau}-\widehat{M_{u_g}}\right) + \frac{\mu_{\psi_0}}{\sigma_{\psi_0}^2}\right\} \times \frac{1}{\sum_g \widehat{M_{\sigma_g^{-2}}}\left(\widehat{M_{b_{1g}}}+\widehat{M_{b_{2g}}}\right) + \frac{1}{\sigma_{\psi_0}^2}}$$

$$
\begin{aligned}
\widehat{M_{u_g}} &\leftarrow \widehat{M_{\sigma_g^{-2}}}\left[\widehat{M_{b_{1g}}}\left(d_g - \widehat{M_\tau} - \widehat{M_\psi}\right) + \widehat{M_{b_{2g}}}\left(d_g - \widehat{M_\tau} + \widehat{M_\psi}\right)\right] \\
&\quad \times \frac{1}{\widehat{M_{\sigma_g^{-2}}}\left(\widehat{M_{b_{1g}}} + \widehat{M_{b_{2g}}}\right) + \widehat{M_{\sigma_\psi^{-2}}}} \\
\widehat{M_{\sigma_\psi^{-2}}} &\leftarrow \frac{A_{\sigma_\psi^2}}{\frac{1}{2}\sum_g \left(\frac{1}{\widehat{M_{\sigma_g^{-2}}}\left(\widehat{M_{b_{1g}}} + \widehat{M_{b_{2g}}}\right) + \widehat{M_{\sigma_\psi^{-2}}}} + \widehat{M_{u_g}}^2\right) + B_\psi} \\
\widehat{M_{\sigma_g^{-2}}} &\leftarrow \frac{A_{\sigma_g^2}}{\widehat{B_{\sigma_g^2}}} \quad \text{where} \\
\widehat{B_{\sigma_g^2}} &\leftarrow \frac{1}{2}\left\{\frac{1}{\sum_g \widehat{M_{\sigma_g^{-2}}} + \frac{1}{\sigma_{\tau_0}^2}} + \left(\widehat{M_{b_{1g}}} + \widehat{M_{b_{2g}}}\right)\left(\frac{1}{\sum_g \widehat{M_{\sigma_g^{-2}}}\left(\widehat{M_{b_{1g}}} + \widehat{M_{b_{2g}}}\right) + \frac{1}{\sigma_{\psi_0}^2}}\right)\right. \\
&\quad + \left(1 - \widehat{M_{b_{1g}}} - \widehat{M_{b_{2g}}}\right)\left(d_g - \widehat{M_\tau}\right)^2 \\
&\quad + \widehat{M_{b_{1g}}}\left[\frac{1}{\widehat{M_{\sigma_g^{-2}}}\left(\widehat{M_{b_{1g}}} + \widehat{M_{b_{2g}}}\right) + \widehat{M_{\sigma_\psi^{-2}}}} + \left(d_g - \widehat{M_\tau} - \widehat{M_\psi} - \widehat{M_{u_g}}\right)^2\right] \\
&\quad \left. + \widehat{M_{b_{2g}}}\left[\frac{1}{\widehat{M_{\sigma_g^{-2}}}\left(\widehat{M_{b_{1g}}} + \widehat{M_{b_{2g}}}\right) + \widehat{M_{\sigma_\psi^{-2}}}} + \left(d_g - \widehat{M_\tau} + \widehat{M_\psi} - \widehat{M_{u_g}}\right)^2\right]\right\} \\
&\quad + \frac{m_g f_g c_g}{2} + B_\varepsilon c_g \\
\widehat{M_{b_{1g}}} &\leftarrow \frac{\exp\left(\widehat{\theta_{1g}}\right)}{\exp\left(\widehat{\theta_{1g}}\right) + \exp\left(\widehat{\theta_{2g}}\right) + 1} \quad \text{and} \\
\widehat{M_{b_{2g}}} &\leftarrow \frac{\exp\left(\widehat{\theta_{2g}}\right)}{\exp\left(\widehat{\theta_{1g}}\right) + \exp\left(\widehat{\theta_{2g}}\right) + 1} \quad \text{where} \\
\widehat{\theta_{1g}} &\leftarrow -\frac{\widehat{M_{\sigma_g^{-2}}}}{2}\left\{\frac{1}{\sum_g \widehat{M_{\sigma_g^{-2}}}\left(\widehat{M_{b_{1g}}} + \widehat{M_{b_{2g}}}\right) + \frac{1}{\sigma_{\psi_0}^2}} + \frac{1}{\widehat{M_{\sigma_g^{-2}}}\left(\widehat{M_{b_{1g}}} + \widehat{M_{b_{2g}}}\right) + \widehat{M_{\sigma_\psi^{-2}}}}\right. \\
&\quad \left. + \left(d_g - \widehat{M_\tau} - \widehat{M_\psi} - \widehat{M_{u_g}}\right)^2 - \left(d_g - \widehat{M_\tau}\right)^2\right\} \\
&\quad + \left[\text{digamma}\left(\sum_g \widehat{M_{b_{1g}}} + \alpha_1\right) - \text{digamma}\left(\sum_g \left(1 - \widehat{M_{b_{1g}}} - \widehat{M_{b_{2g}}}\right) + \alpha_0\right)\right]
\end{aligned}
$$

$$\widehat{\theta_{2g}} \leftarrow -\frac{\widehat{M_{\sigma_g^{-2}}}}{2}\left\{\frac{1}{\sum_g \widehat{M_{\sigma_g^{-2}}}\left(\widehat{M_{b_{1g}}}+\widehat{M_{b_{2g}}}\right)+\frac{1}{\sigma^2_{\psi_0}}}+\frac{1}{\widehat{M_{\sigma_g^{-2}}}\left(\widehat{M_{b_{1g}}}+\widehat{M_{b_{2g}}}\right)+\widehat{M_{\sigma_\psi^{-2}}}}\right.$$
$$\left.+\left(d_g-\widehat{M_\tau}+\widehat{M_\psi}-\widehat{M_{u_g}}\right)^2-\left(d_g-\widehat{M_\tau}\right)^2\right\}$$
$$+\left[\text{digamma}\left(\sum_g \widehat{M_{b_{2g}}}+\alpha_2\right)-\text{digamma}\left(\sum_g\left(1-\widehat{M_{b_{1g}}}-\widehat{M_{b_{2g}}}\right)+\alpha_0\right)\right].$$

iii) Repeat ii) until the increase in

$$\log \underline{p}(\mathbf{y};\mathbf{q}) = \frac{-G}{2}\times\log(2\pi)-\sum_g\left[\widehat{M_{b_{1g}}}\log\widehat{M_{b_{1g}}}+\widehat{M_{b_{2g}}}\log\widehat{M_{b_{2g}}}\right.$$
$$\left.+\left(1-\widehat{M_{b_{1g}}}-\widehat{M_{b_{2g}}}\right)\log\left(1-\widehat{M_{b_{1g}}}-\widehat{M_{b_{1g}}}\right)\right]$$
$$+\log\left(Beta\left(\sum_g\widehat{M_{b_{1g}}}+\alpha_1,\sum_g\widehat{M_{b_{2g}}}+\alpha_2,\sum_g\left(1-\widehat{M_{b_{1g}}}-\widehat{M_{b_{2g}}}\right)+\alpha_0\right)\right)$$
$$-\log\left(Beta\left(\alpha_1,\alpha_2,\alpha_0\right)\right)$$
$$+\left[\log\left(\frac{1}{\sum_g\widehat{M_{\sigma_g^{-2}}}+\frac{1}{\sigma^2_{\tau_0}}}\right)-\log\sigma^2_{\tau_0}+1-\frac{\frac{1}{\sum_g\widehat{M_{\sigma_g^{-2}}}+\frac{1}{\sigma^2_{\tau_0}}}+\left(\widehat{M_\tau}-\mu_{\tau_0}\right)^2}{\sigma^2_{\tau_0}}\right]\times\frac{1}{2}$$
$$+\left[\log\left(\frac{1}{\sum_g\widehat{M_{\sigma_g^{-2}}}\left(\widehat{M_{b_{1g}}}+\widehat{M_{b_{2g}}}\right)+\frac{1}{\sigma^2_{\psi_0}}}\right)-\log\sigma^2_{\psi_0}\right]\times\frac{1}{2}$$
$$+\left[1-\frac{\left(\frac{1}{\sum_g\widehat{M_{\sigma_g^{-2}}}\left(\widehat{M_{b_{1g}}}+\widehat{M_{b_{2g}}}\right)+\frac{1}{\sigma^2_{\psi_0}}}\right)+\left(\widehat{M_\psi}-\mu_{\psi_0}\right)^2}{\sigma^2_{\psi_0}}\right]\times\frac{1}{2}$$
$$+G\times A_\varepsilon\log B_\varepsilon+\sum_g\left[\left(\frac{f_g}{2}+A_\varepsilon\right)\log c_g\right.$$
$$\left.+\log\Gamma\left(A_{\sigma_g^2}\right)-\log\Gamma(A_\varepsilon)-\frac{f_g}{2}\log 2-\log\Gamma\left(\frac{f_g}{2}\right)+\left(\frac{f_g}{2}-1\right)\log m_g+\frac{f_g}{2}\log f_g\right]$$

$$
\begin{aligned}
&+ \sum_g \left\{ \frac{A_{\sigma_g^2}}{\widehat{B_{\sigma_g^2}}} \times \left[ -\frac{1}{2} m_g f_g c_g - B_\varepsilon c_g \right.\right. \\
&- \frac{1}{2}\left( \frac{1}{\sum_g \widehat{M_{\sigma_g^{-2}}} + \frac{1}{\sigma^2_{\tau_0}}} + \left(1 - \widehat{M_{b_{1g}}} - \widehat{M_{b_{2g}}}\right)\left(d_g - \widehat{M_\tau}\right)^2 \right) \\
&- \frac{\widehat{M_{b_{1g}}} + \widehat{M_{b_{1g}}}}{2}\left( \frac{1}{\sum_g \widehat{M_{\sigma_g^{-2}}}\left(\widehat{M_{b_{1g}}} + \widehat{M_{b_{2g}}}\right) + \frac{1}{\sigma^2_{\psi_0}}} \right) \\
&- \frac{\widehat{M_{b_{1g}}}}{2}\left( \frac{1}{\widehat{M_{\sigma_g^{-2}}}\left(\widehat{M_{b_{1g}}} + \widehat{M_{b_{2g}}}\right) + \widehat{M_{\sigma_\psi^{-2}}}} + \left(d_g - \widehat{M_\tau} - \widehat{M_\psi} - \widehat{M_{u_g}}\right)^2 \right) \\
&\left.\left. - \frac{\widehat{M_{b_{2g}}}}{2}\left( \frac{1}{\widehat{M_{\sigma_g^{-2}}}\left(\widehat{M_{b_{1g}}} + \widehat{M_{b_{2g}}}\right) + \widehat{M_{\sigma_\psi^{-2}}}} + \left(d_g - \widehat{M_\tau} + \widehat{M_\psi} - \widehat{M_{u_g}}\right)^2 \right) \right] \right\} \\
&+ \sum_g A_{\sigma_g^2}\left(-\log \widehat{B_{\sigma_g^2}} + 1\right) + \frac{1}{2}\sum_g \left[ \log\left( \frac{1}{\widehat{M_{\sigma_g^{-2}}}\left(\widehat{M_{b_{1g}}} + \widehat{M_{b_{2g}}}\right) + \widehat{M_{\sigma_\psi^{-2}}}} \right) + 1 \right] \\
&+ A_\psi \log B_\psi - \log\Gamma\left(A_\psi\right) + A_{\sigma_\psi^2}\left(-\log \widehat{B_{\sigma_\psi^2}}\right) + \log\Gamma\left(A_{\sigma_\psi^2}\right)
\end{aligned}
$$

from previous iteration becomes negligible.

iv) Upon convergence, the remaining variational parameters are computed:

$$
\begin{aligned}
\widehat{V_\tau} &\leftarrow \frac{1}{\sum_g \widehat{M_{\sigma_g^{-2}}} + \frac{1}{\sigma^2_{\tau_0}}} \\
\widehat{V_\psi} &\leftarrow \frac{1}{\sum_g \widehat{M_{\sigma_g^{-2}}}\left(\widehat{M_{b_{1g}}} + \widehat{M_{b_{2g}}}\right) + \frac{1}{\sigma^2_{\psi_0}}} \\
\widehat{V_{u_g}} &\leftarrow \frac{1}{\widehat{M_{\sigma_g^{-2}}}\left(\widehat{M_{b_{1g}}} + \widehat{M_{b_{2g}}}\right) + \widehat{M_{\sigma_\psi^{-2}}}} \\
\widehat{B_{\sigma_\psi^2}} &\leftarrow \frac{1}{2}\sum_g \left[ \frac{1}{\widehat{M_{\sigma_g^{-2}}}\left(\widehat{M_{b_{1g}}} + \widehat{M_{b_{2g}}}\right) + \widehat{M_{\sigma_\psi^{-2}}}} + \widehat{M_{u_g}}^2 \right] + B_\psi \\
\widehat{\alpha_{p_1}} &\leftarrow \sum_g \widehat{M_{b_{1g}}} + \alpha_1 \\
\widehat{\alpha_{p_2}} &\leftarrow \sum_g \widehat{M_{b_{2g}}} + \alpha_2 \\
\widehat{\alpha_{p_0}} &\leftarrow \sum_g \left(1 - \widehat{M_{b_{1g}}} - \widehat{M_{b_{2g}}}\right) + \alpha_0.
\end{aligned}
$$

## A.2 VB-Proteomics algorithm

### A.2.1 The proteomics model

As pointed out in Section 3.1.3, the fully Bayesian model in Booth et al. (2011) is as follows:

$$
\begin{aligned}
y_{ij}|\mu_{ij} &\sim Poisson(\mu_{ij}) \\
\log \mu_{ij}|I_i, \beta_0, \beta_1, b_{0i}, b_{1i} &= \beta_0 + b_{0i} + \beta_1 T_j + b_{1i} I_i T_j + \beta_2 I_i T_j + \log L_i + \log N_j \\
I_i|\pi_1 &\sim Bernoulli(\pi_1) \quad i.i.d. \\
b_{ki}|\sigma_k^2 &\sim N(0, \sigma_k^2) \quad i.i.d., \quad k = 0, 1 \\
\beta_m &\sim N(0, \sigma_{\beta_m}^2), \quad m = 0, 1, 2 \\
\sigma_k^{-2} &\sim Gamma(A_{\sigma_k^2}, B_{\sigma_k^2}), \quad k = 0, 1 \\
\pi_1 &\sim Beta(\alpha, \beta),
\end{aligned}
$$

where $y_{ij}$ is the spectral count of protein $i$, $i = 1, .., p$, and replicate $j$, $j = 1, .., n$. $L_i$ is the length of protein $i$, $N_j$ is the average count for replicate $j$ over all proteins, and

$$
T_j = \begin{cases} 1 & \text{if replicate } j \text{ is in the treatment group} \\ 0 & \text{if replicate } j \text{ is in the control group.} \end{cases}
$$

In fact, conjugacy in this Poisson GLMM is not sufficient for a tractable solution to be computed by VB. Therefore, a similar Poisson-Gamma HGLM where the parameters $\beta_m, m = 0, 1, 2$ and the latent variables $b_{ki}, k = 0, 1$ are transformed is

used for the VB implementation:

$$
\begin{aligned}
y_{ij}|\mu_{ij} &\sim Poisson(\mu_{ij}) \\
\log \mu_{ij}|I_i, \beta_0, \beta_1, b_{0i}, b_{1i} &= \beta_0 + b_{0i} + \beta_1 T_j + b_{1i} I_i T_j + \beta_2 I_i T_j + \log L_i + \log N_j \\
I_i|\pi_1 &\sim Bernoulli(\pi_1) \quad i.i.d. \\
b_{ki}|\phi_{ki} &= \log(\phi_{ki}^{-1}), \quad k = 0, 1 \\
\phi_{ki}|\delta_k &\sim IG(\delta_k, \delta_k) \quad i.i.d., \quad k = 0, 1 \\
\delta_k &\sim Gamma(A_{\delta_k}, B_{\delta_k}), \quad k = 0, 1 \\
\beta_m|\lambda_m &= \log(\lambda_m^{-1}), \quad m = 0, 1, 2 \\
\lambda_m &\sim IG(A_{\lambda_m}, B_{\lambda_m}), \quad m = 0, 1, 2 \\
\pi_1 &\sim Beta(\alpha, \beta).
\end{aligned}
$$

As before classification is inferred from the posterior expectations of the latent binary indicators $I_i, i = 1, ..., p$.

### A.2.2 Algorithm

The set of observed data and the set of unobserved data are identified as

$$
\begin{aligned}
\mathbf{y} &= \mathbf{y} \\
H &= \{\lambda_0, \lambda_1, \lambda_2, \boldsymbol{\phi_0}, \boldsymbol{\phi_1}, \delta_0, \delta_1, \mathbf{I}, \pi_1\}.
\end{aligned}
$$

The complete likelihood is

$$
\begin{aligned}
p(\mathbf{y}, H) = &\prod_{i,j} p(y_{ij}|\lambda_0, \lambda_1, \lambda_2, \phi_{0i}, \phi_{1i}, I_i) \times \prod_i p(I_i|\pi_1) \times \prod_i p(\phi_{0i}|\delta_0) \times \prod_i p(\phi_{1i}|\delta_1) \\
&\times p(\lambda_0)p(\lambda_1)p(\lambda_2)p(\delta_0)p(\delta_1)p(\pi_1),
\end{aligned}
$$

in which the mixture density is

$$
\begin{aligned}
&p(y_{ij}|\lambda_0, \lambda_1, \lambda_2, \phi_{0i}, \phi_{1i}, I_i) \\
&= Poisson\left(y_{ij};\ \exp(\log \lambda_0^{-1} + \log \phi_{0i}^{-1} + \log(L_i N_j))\right)^{1-T_j} \\
&\times Poisson\left(y_{ij};\ \exp(\log \lambda_0^{-1} + \log \phi_{0i}^{-1} + \log \lambda_1^{-1} + \log(L_i N_j))\right)^{(1-I_i)T_j} \\
&\times Poisson\left(y_{ij};\ \exp(\log \lambda_0^{-1} + \log \phi_{0i}^{-1} + \log \lambda_1^{-1} + \log \phi_{1i}^{-1} + \log \lambda_2^{-1} + \log(L_i N_j))\right)^{I_i T_j}.
\end{aligned}
$$

The log densities that comprise the log complete likelihood are

$$
\begin{aligned}
&\log p(y_{ij}|\lambda_0, \lambda_1, \lambda_2, \phi_{0i}, \phi_{1i}, I_i) \\
&\qquad = (1 - T_j) \times \left[y_{ij}(\log \lambda_0^{-1} + \log \phi_{0i}^{-1}) - \lambda_0^{-1}\phi_{0i}^{-1} L_i N_j\right] \\
&\qquad\quad + (1 - I_i)T_j \times \left[y_{ij}(\log \lambda_0^{-1} + \log \phi_{0i}^{-1} + \log \lambda_1^{-1}) - \lambda_0^{-1}\phi_{0i}^{-1}\lambda_1^{-1} L_i N_j\right] \\
&\qquad\quad + I_i T_j \times \Big[y_{ij}(\log \lambda_0^{-1} + \log \phi_{0i}^{-1} + \log \lambda_1^{-1} + \log \phi_{1i}^{-1} + \log \lambda_2^{-1}) \\
&\qquad\quad -\lambda_0^{-1}\phi_{0i}^{-1}\lambda_1^{-1}\phi_{1i}^{-1}\lambda_2^{-1} L_i N_j\Big] + y_{ij}\log(L_i N_j) - y_{ij}! \\
&\log p(I_i|\pi_1) = I_i \log \pi_1 + (1 - I_i)\log(1 - \pi_1) \\
&\log p(\pi_1) = -\log(Beta(\alpha, \beta)) + (\alpha - 1)\log \pi_1 + (\beta - 1)\log(1 - \pi_1) \\
&\log p(\phi_{ki}|\delta_k) = \delta_k \log \delta_k - \log(\Gamma(\delta_k)) + (\delta_k + 1)\log \phi_k^{-1} - \delta_k \phi_k^{-1}, \quad \text{for } k = 0, 1 \\
&\log p(\delta_k) = -A_{\delta_k}\log B_{\delta_k} - \log(\Gamma(A_{\delta_k})) + (A_{\delta_k} - 1)\log \delta_k - \frac{\delta_k}{B_{\delta_k}}, \quad \text{for } k = 0, 1 \\
&\log p(\lambda_m) = A_{\lambda_m}\log B_{\lambda_m} - \log(\Gamma(A_{\lambda_m})) + (A_{\lambda_m} + 1)\log \lambda_m^{-1} - B_{\lambda_m}\lambda_m^{-1}, \\
&\qquad \text{for } m = 0, 1, 2.
\end{aligned}
$$

The product density restriction

$$q(H) = q_{\lambda_0}(\lambda_0) q_{\lambda_1}(\lambda_1) q_{\lambda_2}(\lambda_2) q_{\boldsymbol{\phi_0}}(\boldsymbol{\phi_0}) q_{\boldsymbol{\phi_1}}(\boldsymbol{\phi_1}) q_{\delta_0}(\delta_0) q_{\delta_1}(\delta_1) q_{\mathbf{I}}(\mathbf{I}) q_{\pi_1}(\pi_1)$$

leads to the following $q$-densities:

- Derivation of $q_{\lambda_0}(\lambda_0)$:

$$
\begin{aligned}
q_{\lambda_0}(\lambda_0) &\propto \exp \mathrm{E}_{-\lambda_0}\{\log p(\mathbf{y}, H)\} \\
&\propto \exp \mathrm{E}_{-\lambda_0}\left\{\sum_{i,j} \log p(y_{ij}|\lambda_0, \lambda_1, \lambda_2, \phi_{0i}, \phi_{1i}, I_i) + \log p(\lambda_0)\right\}
\end{aligned}
$$

$$\propto \exp\Bigg\{\sum_{i,j}\Big[(1-T_j)(y_{ij}\log\lambda_0^{-1}-\lambda_0^{-1}\widehat{M_{\phi_{0i}^{-1}}}L_iN_j)$$
$$+(1-\widehat{M_{I_i}})T_j(y_{ij}\log\lambda_0^{-1}-\lambda_0^{-1}\widehat{M_{\phi_{0i}^{-1}}}\widehat{M_{\lambda_1^{-1}}}L_iN_j)$$
$$+\widehat{M_{I_i}}T_j(y_{ij}\log\lambda_0^{-1}-\lambda_0^{-1}\widehat{M_{\phi_{0i}^{-1}}}\widehat{M_{\lambda_1^{-1}}}\widehat{M_{\phi_{1i}^{-1}}}\widehat{M_{\lambda_2^{-1}}}L_iN_j)\Big]$$
$$+(A_{\lambda_0}+1)\log\lambda_0^{-1}-\frac{B_{\lambda_0}}{\lambda_0}\Bigg\}$$

The kernel of an Inverse-Gamma density is identified on the right hand side. Therefore, it can be deduced that

$$q_{\lambda_0}(\lambda_0) = IG(\widehat{A_{\lambda_0}}, \widehat{B_{\lambda_0}})$$

with

$$\widehat{A_{\lambda_0}} = \sum_{i,j} y_{ij} + A_{\lambda_0}$$
$$\widehat{B_{\lambda_0}} = \sum_{i,j} L_iN_j\Big[(1-T_j)\widehat{M_{\phi_{0i}^{-1}}} + (1-\widehat{M_{I_i}})T_j\widehat{M_{\phi_{0i}^{-1}}}\widehat{M_{\lambda_1^{-1}}}$$
$$+\widehat{M_{I_i}}T_j\widehat{M_{\phi_{0i}^{-1}}}\widehat{M_{\lambda_1^{-1}}}\widehat{M_{\phi_{1i}^{-1}}}\widehat{M_{\lambda_2^{-1}}}\Big] + B_{\lambda_0}.$$

Moreover, the posterior mean and posterior expected log of $\lambda_0^{-1}$ are

$$\widehat{M_{\lambda_0^{-1}}} = \frac{\widehat{A_{\lambda_0}}}{\widehat{B_{\lambda_0}}}$$
$$\widehat{\log\lambda_0^{-1}} = \text{digamma}(\widehat{A_{\lambda_0}}) - \log\widehat{B_{\lambda_0}}.$$

- Derivation of $q_{\lambda_1}(\lambda_1)$ and $q_{\lambda_2}(\lambda_2)$ is similar to that of $q_{\lambda_0}(\lambda_0)$:

$$q(\lambda_1) \propto \exp \mathrm{E}_{-\lambda_1}\left\{\sum_{i,j}\log p(y_{ij}|\lambda_0,\lambda_1,\lambda_2,\phi_{0i},\phi_{1i},I_i) + \log p(\lambda_1)\right\}$$

$$\Rightarrow q(\lambda_1) = IG(\widehat{A_{\lambda_1}}, \widehat{B_{\lambda_1}})$$
$$\widehat{A_{\lambda_1}} = \sum_{i,j} T_j y_{ij} + A_{\lambda_1}$$
$$\widehat{B_{\lambda_1}} = \sum_{i,j} L_iN_j[(1-\widehat{M_{I_i}})T_j\widehat{M_{\lambda_0^{-1}}}\widehat{M_{\phi_{0i}^{-1}}} + \widehat{M_{I_i}}T_j\widehat{M_{\lambda_0^{-1}}}\widehat{M_{\phi_{0i}^{-1}}}\widehat{M_{\phi_{1i}^{-1}}}\widehat{M_{\lambda_2^{-1}}}]$$
$$+ B_{\lambda_1},$$

and

$$\widehat{M_{\lambda_1^{-1}}} = \frac{\widehat{A_{\lambda_1}}}{\widehat{B_{\lambda_1}}}$$

$$\widehat{\log \lambda_1^{-1}} = \text{digamma}(\widehat{A_{\lambda_1}}) - \log \widehat{B_{\lambda_1}}.$$

$$q_{\lambda_2}(\lambda_2) \propto \exp \mathrm{E}_{-\lambda_2} \left\{ \sum_{i,j} \log p(y_{ij}|\lambda_0, \lambda_1, \lambda_2, \phi_{0i}, \phi_{1i}, I_i) + \log p(\lambda_2) \right\}$$

$$\Rightarrow q_{\lambda_2}(\lambda_2) = IG(\widehat{A_{\lambda_2}}, \widehat{B_{\lambda_2}})$$

$$\widehat{A_{\lambda_2}} = \sum_{i,j} \widehat{M_{I_i}} T_j y_{ij} + A_{\lambda_2}$$

$$\widehat{B_{\lambda_2}} = \sum_{i,j} L_i N_j [\widehat{M_{I_i}} T_j \widehat{M_{\lambda_0^{-1}}} \widehat{M_{\phi_{0i}^{-1}}} \widehat{M_{\lambda_1^{-1}}} \widehat{M_{\phi_{1i}^{-1}}}] + B_{\lambda_2},$$

and

$$\widehat{M_{\lambda_2^{-1}}} = \frac{\widehat{A_{\lambda_2}}}{\widehat{B_{\lambda_2}}}$$

$$\widehat{\log \lambda_2^{-1}} = \text{digamma}(\widehat{A_{\lambda_2}}) - \log \widehat{B_{\lambda_2}}.$$

- Derivation of $q_{\boldsymbol{\phi_0}}(\boldsymbol{\phi_0})$ and $q_{\boldsymbol{\phi_1}}(\boldsymbol{\phi_1})$ is also similar to that of $q_{\lambda_0}(\lambda_0)$, with induced factorizations:

$$q_{\boldsymbol{\phi_0}}(\boldsymbol{\phi_0}) \propto \exp \mathrm{E}_{-\boldsymbol{\phi_0}} \left\{ \sum_{i,j} \log p(y_{ij}|\lambda_0, \lambda_1, \lambda_2, \phi_{0i}, \phi_{1i}, I_i) + \sum_i \log p(\phi_{0i}|\delta_0) \right\}$$

$$\Rightarrow \quad q_{\boldsymbol{\phi_0}}(\boldsymbol{\phi_0}) = \prod_i q_{\phi_{0i}}(\phi_{0i}) \text{ and}$$

$$q_{\phi_{0i}}(\phi_{0i}) \propto \exp \mathrm{E}_{-\boldsymbol{\phi_0}} \left\{ \sum_j \log p(y_{ij}|\lambda_0, \lambda_1, \lambda_2, \phi_{0i}, \phi_{1i}, I_i) + \log p(\phi_{0i}|\delta_0) \right\}.$$

Therefore, for each $i$,

$$q_{\phi_{0i}}(\phi_{0i}) = IG(\widehat{A_{\phi_{0i}}}, \widehat{B_{\phi_{0i}}})$$

$$\widehat{A_{\phi_{0i}}} = \sum_j y_{ij} + \widehat{M_{\delta_0}}$$

$$\widehat{B_{\phi_{0i}}} = \sum_j L_i N_j \widehat{M_{\lambda_0^{-1}}} [(1 - T_j) + (1 - \widehat{M_{I_i}}) T_j \widehat{M_{\lambda_1^{-1}}}$$

$$+ \widehat{M_{I_i}} T_j \widehat{M_{\lambda_1^{-1}}} \widehat{M_{\phi_{1i}^{-1}}} \widehat{M_{\lambda_2^{-1}}}] + \widehat{M_{\delta_0}}$$

$$\widehat{M_{\phi_{0i}}^{-1}} = \frac{\widehat{A_{\phi_{0i}}}}{\widehat{B_{\phi_{0i}}}}$$

$$\log \widehat{\phi_{0i}}^{-1} = \text{digamma}(\widehat{A_{\phi_{0i}}}) - \log \widehat{B_{\phi_{0i}}}.$$

$$q_{\boldsymbol{\phi_1}}(\boldsymbol{\phi_1}) \propto \exp \mathrm{E}_{-\boldsymbol{\phi_1}} \left\{ \sum_{i,j} \log p(y_{ij}|\lambda_0, \lambda_1, \lambda_2, \phi_{0i}, \phi_{1i}, I_i) + \sum_i \log p(\phi_{1i}|\delta_1) \right\}$$

$$\Rightarrow \quad q_{\boldsymbol{\phi_1}}(\boldsymbol{\phi_1}) = \prod_i q_{\phi_{1i}}(\phi_{1i}) \text{ and}$$

$$q_{\phi_{1i}}(\phi_{1i}) \propto \exp \mathrm{E}_{-\boldsymbol{\phi_1}} \left\{ \sum_j \log p(y_{ij}|\lambda_0, \lambda_1, \lambda_2, \phi_{0i}, \phi_{1i}, I_i) + \log p(\phi_{1i}|\delta_1) \right\}.$$

Therefore, for each $i$,

$$q_{\phi_{1i}}(\phi_{1i}) = IG(\widehat{A_{\phi_{1i}}}, \widehat{B_{\phi_{1i}}})$$

$$\widehat{A_{\phi_{1i}}} = \sum_j \widehat{M_{I_i}} T_j y_{ij} + \widehat{M_{\delta_1}}$$

$$\widehat{B_{\phi_{1i}}} = \sum_j L_i N_j \widehat{M_{I_i}} T_j \widehat{M_{\lambda_0^{-1}}} \widehat{M_{\phi_{0i}^{-1}}} \widehat{M_{\lambda_1^{-1}}} \widehat{M_{\lambda_2^{-1}}} + \widehat{M_{\delta_1}}$$

$$\widehat{M_{\phi_{1i}^{-1}}} = \frac{\widehat{A_{\phi_{1i}}}}{\widehat{B_{\phi_{1i}}}}$$

$$\log \widehat{\phi_{1i}}^{-1} = \text{digamma}(\widehat{A_{\phi_{1i}}}) - \log \widehat{B_{\phi_{1i}}}.$$

- Derivation of $q_{\delta_0}(\delta_0)$:

$$q_{\delta_0}(\delta_0) \propto \exp \mathrm{E}_{-\delta_0} \left\{ \sum_i \log p(\phi_{0i}|\delta_0) + \log p(\delta_0) \right\}$$

$$\propto \exp \left\{ \sum_i \left[ \delta_0 \log \delta_0 - \log \Gamma(\delta_0) + (\delta_0 + 1) \widehat{\log \phi_{0i}^{-1}} - \delta_0 \widehat{M_{\phi_{0i}^{-1}}} \right] \right.$$

$$\left. + - A_{\delta_0} \log B_{\delta_0} - \log \Gamma(A_{\delta_0}) + (A_{\delta_0} - 1) \log \delta_0 - \frac{\delta_0}{B_{\delta_0}} \right\}.$$

The right hand side does not contain the kernel of any standard distribution. Therefore, an approximation to $\log \Gamma(\delta_0)$ (Rocktaeschel, 1922) is used:

For complex number z with large $Re(z)$, because $\Gamma(z+1) = z! = z\Gamma(z)$,

$$\begin{aligned}
\log\Gamma(z) &= \log\Gamma(z+1) - \log z \\
&= \log z! - \log z \\
&\approx \left(\frac{1}{2}\log(2\pi z) + z\log z - z\right) - \log z \\
&\quad \text{by Stirling's approximation } n! \approx \sqrt{2\pi n}\left(\frac{n}{e}\right)^n \\
&\approx (z - \frac{1}{2})\log z - z + \frac{1}{2}\log(2\pi).
\end{aligned}$$

Hence,

$$\delta_0 \log\delta_0 - \log\Gamma(\delta_0) \approx \frac{1}{2}\log\delta_0 + \delta_0 - \frac{1}{2}\log(2\pi) \text{ for large } \delta_0 > 0.$$

Substituting the above on the right hand side of the formula for $q_{\delta_0}(\delta_0)$ leads to the kernel of a Gamma density. Therefore,

$$\begin{aligned}
q_{\delta_0}(\delta_0) &\approx Gamma(\widehat{A_{\delta_0}}, \widehat{B_{\delta_0}}) \\
\widehat{A_{\delta_0}} &= \frac{p}{2} + A_{\delta_0} \\
\widehat{B_{\delta_0}} &= \frac{1}{-p - \sum_i \widehat{\log\phi_{0i}^{-1}} + \sum_i \widehat{M_{\phi_{0i}^{-1}}} + \frac{1}{B_{\delta_0}}},
\end{aligned}$$

and

$$\widehat{M_{\delta_0}} = \widehat{A_{\delta_0}}\widehat{B_{\delta_0}}.$$

- Derivation of $q_{\delta_1}(\delta_1)$ is similar to that of $q_{\delta_0}(\delta_0)$, with the same approximation employed:

$$\begin{aligned}
q_{\delta_1}(\delta_1) &\approx Gamma(\widehat{A_{\delta_1}}, \widehat{B_{\delta_1}}) \\
\widehat{A_{\delta_1}} &= \frac{p}{2} + A_{\delta_1} \\
\widehat{B_{\delta_1}} &= \frac{1}{-p - \sum_i \widehat{\log\phi_{1i}^{-1}} + \sum_i \widehat{M_{\phi_{1i}^{-1}}} + \frac{1}{B_{\delta_1}}},
\end{aligned}$$

and

$$\widehat{M_{\delta_1}} = \widehat{A_{\delta_1}}\widehat{B_{\delta_1}}.$$

- Derivation of $q_{\mathbf{I}}(\mathbf{I})$:

$$q_{\mathbf{I}}(\mathbf{I}) \propto \exp \mathrm{E}_{-\mathbf{I}} \left\{ \sum_{i,j} \log p(y_{ij}|\lambda_0, \lambda_1, \lambda_2, \phi_{0i}, \phi_{1i}, I_i) + \sum_i \log p(I_i|\pi_1) \right\}$$

$$\Rightarrow \quad q_{\mathbf{I}}(\mathbf{I}) = \prod_i q_{I_i}(I_i) \text{ and}$$

$$q_{I_i}(I_i) \propto \exp \mathrm{E}_{-\mathbf{I}} \left\{ \sum_j \log p(y_{ij}|\lambda_0, \lambda_1, \lambda_2, \phi_{0i}, \phi_{1i}, I_i) + \log p(I_i|\pi_1) \right\}.$$

Therefore, for each $i$,

$$q_{I_i}(I_i) = \mathit{Bernoulli}\left( \frac{\exp(\widehat{\theta_i})}{\exp(\widehat{\theta_i}) + 1} \right)$$

$$\widehat{\theta_i} = \sum_j y_{ij} T_j \left( \widehat{\log \phi_{1i}^{-1}} + \widehat{\log \lambda_2^{-1}} \right)$$

$$+ L_i \widehat{M_{\phi_{0i}^{-1}}} \left( \sum_j N_j T_j \right) \widehat{M_{\lambda_0^{-1}}} \widehat{M_{\lambda_1^{-1}}} \left( 1 - \widehat{M_{\phi_{1i}^{-1}}} \widehat{M_{\lambda_2^{-1}}} \right)$$

$$+ \widehat{\log \pi_1} - \widehat{\log(1 - \pi_1)},$$

and

$$\widehat{M_{I_i}} = \frac{\exp(\widehat{\theta_i})}{\exp(\widehat{\theta_i}) + 1}.$$

- Derivation of $q_{\pi_1}(\pi_1)$:

$$q_{\pi_1}(\pi_1) \propto \exp \mathrm{E}_{-\pi_1} \left\{ \sum_i \log p(I_i|\pi_1) + \log p(\pi_1) \right\}$$

$$\Rightarrow q_{\pi_1}(\pi_1) = \mathit{Beta}(\widehat{\alpha_{\pi_1}}, \widehat{\beta_{\pi_1}})$$

$$\widehat{\alpha_{\pi_1}} = \sum_i \widehat{M_{I_i}} + \alpha$$

$$\widehat{\beta_{\pi_1}} = \sum_i (1 - \widehat{M_{I_i}}) + \beta,$$

and

$$\widehat{\log \pi_1} = \mathrm{digamma}(\widehat{\alpha_{\pi_1}}) - \mathrm{digamma}(\widehat{\alpha_{\pi_1}} + \widehat{\beta_{\pi_1}})$$

$$\widehat{\log(1 - \pi_1)} = \mathrm{digamma}(\widehat{\beta_{\pi_1}}) - \mathrm{digamma}(\widehat{\alpha_{\pi_1}} + \widehat{\beta_{\pi_1}})$$

Updating the $q$-densities in an iterative scheme boils down to updating the variational parameters in the scheme. Convergence is monitored via the scalar quantity $C_q(\mathbf{y})$, the lower bound on the log of the marginal data density:

$$
\begin{aligned}
\log \underline{p}\,(\mathbf{y};\mathbf{q}) &= \mathrm{E}_H \log p(\mathbf{y}, H) - \mathrm{E}_H \log q(H) \\
&= \mathrm{E}_H \left\{ \sum_{i,j} \log p(y_{ij}|\lambda_0, \lambda_1, \lambda_2, \phi_{0i}, \phi_{1i}, I_i) - \sum_i \log q_{I_i}(I_i) \right\} \\
&\quad + \mathrm{E}_H \left\{ \sum_i \log p(I_i|\pi_1) + \log p(\pi_1) - \log q_{\pi_1}(\pi_1) \right\} \\
&\quad + \mathrm{E}_H \left\{ \log p(\lambda_0) + \log p(\lambda_1) + \log p(\lambda_2) - \log q_{\lambda_0}(\lambda_0) - \log q_{\lambda_1}(\lambda_1) - \log q_{\lambda_2}(\lambda_2) \right\} \\
&\quad + \mathrm{E}_H \left\{ \sum_i \log p(\phi_{0i}|\delta_0) + \sum_i \log p(\phi_{1i}|\delta_1) + \log p(\delta_0) + \log p(\delta_1) \right\} \\
&\quad + \mathrm{E}_H \left\{ - \sum_i \log q_{\phi_{0i}}(\phi_{0i}) - \sum_i \log q_{\phi_{1i}}(\phi_{1i}) - \log q_{\delta_0}(\delta_0) - \log q_{\delta_1}(\delta_1) \right\} \\
&= \sum_{i,j} \left\{ (1 - T_j) \left[ y_{ij} \left( \widehat{\log \lambda_0^{-1}} + \widehat{\log \phi_{0i}^{-1}} + \log(L_i N_j) \right) - \widehat{M_{\lambda_0^{-1}}} \widehat{M_{\phi_{0i}^{-1}}} L_i N_j \right] \right\} \\
&\quad + \sum_{i,j} \left\{ (1 - \widehat{M_{I_i}}) T_j \left[ y_{ij} \left( \widehat{\log \lambda_0^{-1}} + \widehat{\log \phi_{0i}^{-1}} + \widehat{\log \lambda_1^{-1}} + \log(L_i N_j) \right) \right. \right. \\
&\quad \left. \left. - \widehat{M_{\lambda_0^{-1}}} \widehat{M_{\phi_{0i}^{-1}}} \widehat{M_{\lambda_1^{-1}}} L_i N_j \right] \right\} \\
&\quad + \sum_{i,j} \left\{ \widehat{M_{I_i}} T_j \left[ y_{ij} \left( \widehat{\log \lambda_0^{-1}} + \widehat{\log \phi_{0i}^{-1}} + \widehat{\log \lambda_1^{-1}} + \widehat{\log \phi_{1i}^{-1}} + \widehat{\log \lambda_2^{-1}} + \log(L_i N_j) \right) \right. \right. \\
&\quad \left. \left. - \widehat{M_{\lambda_0^{-1}}} \widehat{M_{\phi_{0i}^{-1}}} \widehat{M_{\lambda_1^{-1}}} \widehat{M_{\phi_{1i}^{-1}}} \widehat{M_{\lambda_2^{-1}}} L_i N_j \right] \right\} + \sum_{i,j} \left\{ -y_{ij}! \right\} \\
&\quad - \sum_i \left( \widehat{M_{I_i}} \log \widehat{M_{I_i}} + (1 - \widehat{M_{I_i}}) \log(1 - \widehat{M_{I_i}}) \right) \\
&\quad + \left[ -\log(Beta(\alpha, \beta)) + \log(Beta(\widehat{\alpha_{\pi_1}}, \widehat{\beta_{\pi_1}})) \right] \\
&\quad + A_{\lambda_0} \log B_{\lambda_0} - \log \Gamma(A_{\lambda_0}) - \widehat{A_{\lambda_0}} \log \widehat{B_{\lambda_0}} + \log \Gamma(\widehat{A_{\lambda_0}}) + \widehat{A_{\lambda_0}} \\
&\quad - \left( \sum_{i,j} y_{ij} \right) \widehat{\log \lambda_0^{-1}} - B_{\lambda_0} \widehat{M_{\lambda_0^{-1}}} \\
&\quad + A_{\lambda_1} \log B_{\lambda_1} - \log \Gamma(A_{\lambda_1}) - \widehat{A_{\lambda_1}} \log \widehat{B_{\lambda_1}} + \log \Gamma(\widehat{A_{\lambda_1}}) + \widehat{A_{\lambda_1}} \\
&\quad - \left( \sum_{i,j} T_j y_{ij} \right) \widehat{\log \lambda_1^{-1}} - B_{\lambda_1} \widehat{M_{\lambda_1^{-1}}} \\
&\quad + A_{\lambda_2} \log B_{\lambda_2} - \log \Gamma(A_{\lambda_2}) - \widehat{A_{\lambda_2}} \log \widehat{B_{\lambda_2}} + \log \Gamma(\widehat{A_{\lambda_2}}) + \widehat{A_{\lambda_2}} \\
&\quad - \left( \sum_{i,j} \widehat{M_{I_i}} T_j y_{ij} \right) \widehat{\log \lambda_2^{-1}} - B_{\lambda_2} \widehat{M_{\lambda_2^{-1}}}
\end{aligned}
$$

$$+\sum_{k=0,1}\left\{\sum_i(\widehat{M_{\delta_k}}-\widehat{A_{\phi_{ki}}})\log\phi_{ki}^{-1}-\widehat{M_{\delta_k}}\left(\sum_i\widehat{M_{\phi_{ki}^{-1}}}-p\right)-\frac{p}{2}\log(2\pi)\right.$$
$$\left.-\sum_i\left(\widehat{A_{\phi_{ki}}}\log\widehat{B_{\phi_{ki}}}-\log\Gamma(\widehat{A_{\phi_{ki}}})-\widehat{A_{\phi_{ki}}}\right)\right\}$$
$$+\sum_{k=0,1}\left\{-A_{\delta_k}\log B_{\delta_k}-\log(\Gamma(A_{\delta_k}))-\frac{\widehat{M_{\delta_k}}}{B_{\delta_k}}+\widehat{A_{\delta_k}}\log\widehat{B_{\delta_k}}+\log(\Gamma(\widehat{A_{\delta_k}}))+\widehat{A_{\delta_k}}\right\}.$$

The VB-proteomics algorithm consists of the following steps:

Step 1: Initialize $\widehat{B_{\lambda_0}}, \widehat{B_{\lambda_1}}, \widehat{B_{\delta_0}}, \widehat{B_{\delta_1}}$, and $\widehat{A_{\phi_{0i}}}, \widehat{A_{\phi_{1i}}}, \widehat{B_{\phi_{0i}}}, \widehat{B_{\phi_{1i}}}$ for each $i$.

Step 2: Cycle through $\widehat{A_{\lambda_2}}, \widehat{B_{\lambda_2}}, \widehat{B_{\lambda_0}}, \widehat{B_{\lambda_1}}, \widehat{B_{\delta_0}}, \widehat{B_{\delta_1}}, \widehat{A_{\phi_{0i}}}, \widehat{B_{\phi_{0i}}}, \widehat{A_{\phi_{1i}}}, \widehat{B_{\phi_{1i}}}, \widehat{M_{I_i}}$ iteratively, until the increase in $C_q(\mathbf{y})$ evaluated at the end of each iteration is negligible.

Step 3: Compute $\widehat{\alpha_{\pi_1}}$ and $\widehat{\beta_{\pi_1}}$ using converged variational parameter values.

The values of the model parameters were chosen to form non-informative priors: shape and scale parameters for the Inverse-Gamma priors were set to be 0.1, shape and scale parameters for the Gamma priors were set to be 0.1 and 100 respectively, and parameters for the Beta prior were both 1. Because a log-normal distribution is approximately a Gamma distribution, variance of $e^{b_{ki}}, k = 0, 1$ which follows a log-normal distribution in the Poisson GLMM roughly equals variance of $\phi_{ki}^{-1}, k = 0, 1$ which follows a Gamma distribution in the Poisson-Gamma HGLM. That is, $(e^{\sigma_k^2} - 1)e^{\sigma_k^2} \approx 1/\delta_k,\ k = 0, 1$. Based on this we determined parameter values in the $Gamma(A_{\delta_k}, B_{\delta_k})$ prior for $\delta_k,\ k = 0, 1$. Starting values of posterior mean of the latent indicator were $\widehat{M_{I_i}} = 1$ for proteins that are associated with the 20% smallest p-values from one protein at a time Score tests and $\widehat{M_{I_i}} = 0$ otherwise.

An approximation to the digamma function, digamma$(z) \approx \log z - \frac{1}{2z}$, was used (Bernardo, J. M., 1976) wherever $z$ was too small in the VB implementation.

APPENDIX B

**TECHNICAL DETAILS OF CHAPTER 4**

## B.1 RRmix model and EM algorithm

The RRmix model is illustrated per gene as follows.

$$
\begin{aligned}
y_g &= \mu + X_c\gamma_g + X\beta_g + \Lambda F_g + W_g \qquad \text{(B.1)}\\
W_g &\sim N_n(0, \sigma_g^2 I_n)\\
F_g &\sim N_q(0, I_q)\\
\beta_g \big| b_g &\sim N_2\left(\begin{bmatrix} 0 \\ b_g\psi \end{bmatrix},\ (1-b_g)\begin{bmatrix} \sigma_0^2 & 0 \\ 0 & 0 \end{bmatrix} + b_g\begin{bmatrix} \sigma_0^2 & 0 \\ 0 & \sigma_1^2 \end{bmatrix}\right)\\
b_g &\sim Bernoulli(p)\\
\sigma_g^2 &\sim IG(A, B),
\end{aligned}
$$

where $y_g, \beta_g, F_g, W_g$ are the $g$-th column in the $n \times G$ matrix $Y$, $2 \times G$ matrix $\boldsymbol{\beta}$, $q \times G$ matrix $\mathbf{F}$, and $n \times G$ matrix $\mathbf{W}$ respectively, and $\underset{n\times 2}{X} = [\mathbf{1}_n, \mathbf{x}]$ with $\mathbf{x}$ representing the given SNP vector, $\underset{2\times 1}{\beta_g} = \begin{bmatrix} \beta_{0g} \\ \beta_{1g} \end{bmatrix}$, and $\underset{n\times 4}{X_c} = [\mathbf{gen}, \mathbf{dis}, \mathbf{smo}, \mathbf{ld1}]$. $X_c$ is the matrix for the four fixed covariates: gender (**gen**), disease status (**dis**), smoking status (**smo**), and loading of first factor (**ld1**) from factor analysis on SNP covariance matrix. Loading of first factor from factor analysis on SNP covariance matrix is included to account for population structure, which is employed in Gao et al. (2014). $IG(A, B)$ denotes an Inverse-Gamma distribution with shape $A$ and scale $B$.

The RRmix model (B.1) is equivalent to

$$
\begin{aligned}
y_g &= \mu + X_c\gamma_g + X\beta_g + \mathbf{x}(b_g\psi) + \Lambda F_g + W_g \qquad \text{(B.2)}\\
W_g &\sim N_n(0, \sigma_g^2 I_n)\\
F_g &\sim N_q(0, I_q)\\
\beta_g \big| b_g &\sim N_2\left( 0\ ,\ (1-b_g)\begin{bmatrix}\sigma_0^2 & 0\\ 0 & 0\end{bmatrix} + b_g\begin{bmatrix}\sigma_0^2 & 0\\ 0 & \sigma_1^2\end{bmatrix}\right)\\
b_g &\sim Bernoulli(p)\\
\sigma_g^2 &\sim IG(A, B).
\end{aligned}
$$

This is because, the conditional distribution of vector $\beta_g$ given $b_g$ implies that

$$
\begin{aligned}
\beta_{0g} &\sim N(0, \sigma_0^2)\\
\beta_{1g} \big| b_g = 0 &\equiv 0\\
\beta_{1g} \big| b_g = 1 &\sim N\left(\psi, \sigma_1^2\right),
\end{aligned}
$$

and $\beta_g \big| b_g$ follows $(1-b_g)N_2\left( 0\ ,\ \begin{bmatrix}\sigma_0^2 & 0\\ 0 & 0\end{bmatrix}\right) + b_g N_2\left( \begin{bmatrix}0\\ \psi\end{bmatrix}\ ,\ \begin{bmatrix}\sigma_0^2 & 0\\ 0 & \sigma_1^2\end{bmatrix}\right)$, a mixture distribution, which can be written as $(1-b_g)N_2\left( 0\ ,\ \begin{bmatrix}\sigma_0^2 & 0\\ 0 & 0\end{bmatrix}\right) + b_g N_2\left( 0\ ,\ \begin{bmatrix}\sigma_0^2 & 0\\ 0 & \sigma_1^2\end{bmatrix}\right) + b_g\psi$.

We derive EM updates based on the RRmix model (B.2). For a single SNP, the latent indicator $b_g$ tells whether gene $g$ is having null or non-null gene-SNP association via $\beta_{1g}$. $\beta_{1g}$ for each gene is estimated while controlling for the sample means via $\mu$, the main effect of gene $g$ via $\beta_{0g}$, covariate effects $\gamma_g$, and hidden effects $\Lambda F_g$.

We apply EM algorithm to estimate $\mu, \sigma_0^2, \sigma_1^2, \Lambda, p, \gamma_g, \psi$ with missing data $\{\beta_g\}, \{F_g\}, \{b_g\}$ , and shrinkage estimation is done for $\{\sigma_g^2\}$ which is assigned prior

*Inverse-Gamma*$(A, B)$ with unknown hyperparameters $A, B$.

The complete data log likelihood is $l = \sum_g l_g$ where $l_g$ is the complete data log likelihood for gene $g$

$$\begin{aligned} l_g &= \log p(y_g|\beta_g, F_g, b_g; \mu, \Lambda, \psi, \gamma_g, \sigma_g^2) + \log p(\beta_g|b_g; \sigma_0^2, \sigma_1^2) \\ &\quad + \log p(F_g) + \log p(b_g; p) \\ &= -\frac{n}{2}\log\sigma_g^2 - \frac{1}{2\sigma_g^2}\left\|y_g - \mu - X_c\gamma_g - \mathbf{x}(b_g\psi) - X\beta_g - \Lambda F_g\right\|^2 \\ &\quad - \frac{1}{2}\log\left|(1-b_g)\begin{bmatrix}\sigma_0^2 & 0\\ 0 & 0\end{bmatrix} + b_g\begin{bmatrix}\sigma_0^2 & 0\\ 0 & \sigma_1^2\end{bmatrix}\right| \\ &\quad - \frac{1}{2}\beta_g^T\left[(1-b_g)\begin{bmatrix}\sigma_0^2 & 0\\ 0 & 0\end{bmatrix} + b_g\begin{bmatrix}\sigma_0^2 & 0\\ 0 & \sigma_1^2\end{bmatrix}\right]^{-1}\beta_g \\ &\quad - \frac{1}{2}\left\|F_g\right\|^2 + b_g\log p + (1-b_g)\log(1-p), \end{aligned} \tag{B.3}$$

apart from an additive constant. For notational convenience in subsequent derivations, let $\underset{2\times2}{A_0} = \begin{bmatrix}1 & 0\\ 0 & 0\end{bmatrix}$ and $\underset{2\times2}{\Psi_\beta} = \begin{bmatrix}\sigma_0^2 & 0\\ 0 & \sigma_1^2\end{bmatrix}$, so that $A_0\Psi_\beta = \begin{bmatrix}\sigma_0^2 & 0\\ 0 & 0\end{bmatrix}$ hence $(1-b_g)A_0\Psi_\beta + b_g\Psi_\beta = (1-b_g)\begin{bmatrix}\sigma_0^2 & 0\\ 0 & 0\end{bmatrix} + b_g\begin{bmatrix}\sigma_0^2 & 0\\ 0 & \sigma_1^2\end{bmatrix}$ for any given $g$.

### B.1.1 E-step

Model (B.2) implies the following $(2 + q + n)$-dimensional multivariate normal distribution

$$\left[\begin{bmatrix}\beta_g\\ F_g\end{bmatrix}\right] \Bigg| \, b_g \sim N\left(\begin{bmatrix}0_{2+q}\\ \mu + X_c\gamma_g + \mathbf{x}(b_g\psi)\end{bmatrix}, \quad T_g\right) \tag{B.4}$$

where

$$T_g = \begin{bmatrix} \begin{bmatrix} ((1-b_g)A_0\Psi_\beta + b_g\Psi_\beta) & 0 \\ 0 & I_q \end{bmatrix} & \begin{bmatrix} ((1-b_g)A_0\Psi_\beta + b_g\Psi_\beta)X^T \\ \Lambda^T \end{bmatrix} \\ \begin{bmatrix} X((1-b_g)A_0\Psi_\beta + b_g\Psi_\beta)\ ,\ \Lambda \end{bmatrix} & X((1-b_g)A_0\Psi_\beta + b_g\Psi_\beta)X^T + \Lambda\Lambda^T + \sigma_g^2 I_n \end{bmatrix}.$$

Hence, the conditional distribution $\begin{bmatrix} \beta_g \\ F_g \end{bmatrix} \Big| y_g, b_g$ has mean vector and covariance matrix

$$\begin{bmatrix} \hat{\beta}_g \\ \hat{F}_g \end{bmatrix} = \mathbb{E}\left( \begin{bmatrix} \beta_g \\ F_g \end{bmatrix} \Big| y_g, b_g \right) = T_{g12}T_{g22}^{-1}\left(y_g - \mu - X_c\gamma_g - \mathbf{x}(b_g\psi)\right)$$

$$\hat{\Sigma}_g = \mathbb{V}\left( \begin{bmatrix} \beta_g \\ F_g \end{bmatrix} \Big| y_g, b_g \right) = T_{g11} - T_{g12}T_{g22}^{-1}T_{g21}$$

where $T_g$ is partitioned into four parts

$$T_{g11} = \begin{bmatrix} ((1-b_g)A_0\Psi_\beta + b_g\Psi_\beta) & 0 \\ 0 & I_q \end{bmatrix}$$

$$T_{g12} = \begin{bmatrix} ((1-b_g)A_0\Psi_\beta + b_g\Psi_\beta)X^T \\ \Lambda^T \end{bmatrix}$$

$$T_{g21} = \begin{bmatrix} X((1-b_g)A_0\Psi_\beta + b_g\Psi_\beta)\ ,\ \Lambda \end{bmatrix}$$

$$T_{g22} = X((1-b_g)A_0\Psi_\beta + b_g\Psi_\beta)X^T + \Lambda\Lambda^T + \sigma_g^2 I_n\,.$$

We further represent $\hat{\Sigma}_g$ as

$$\hat{\Sigma}_g = \begin{bmatrix} \hat{\Sigma}_{g11} & \hat{\Sigma}_{g12} \\ \hat{\Sigma}_{g21} & \hat{\Sigma}_{g22} \end{bmatrix},$$

where the block matrices are of dimensions $2 \times 2$, $2 \times q$, $q \times 2$, $q \times q$, respectively. They are

$$\hat{\Sigma}_{g11} = ((1 - b_g)A_0\Psi_\beta + b_g\Psi_\beta) \left[I_2 - X^T \left[X((1 - b_g)A_0\Psi_\beta + b_g\Psi_\beta)X^T + \Lambda\Lambda^T + \sigma_g^2 I_n\right]^{-1} X((1 - b_g)A_0\Psi_\beta + b_g\Psi_\beta)\right]$$

$$\hat{\Sigma}_{g12} = \hat{\Sigma}_{g21}^T$$

$$\hat{\Sigma}_{g21} = -\Lambda^T \left[X((1 - b_g)A_0\Psi_\beta + b_g\Psi_\beta)X^T + \Lambda\Lambda^T + \sigma_g^2 I_n\right]^{-1} X((1 - b_g)A_0\Psi_\beta + b_g\Psi_\beta)$$

$$\hat{\Sigma}_{g22} = I_q - \Lambda^T \left[X((1 - b_g)A_0\Psi_\beta + b_g\Psi_\beta)X^T + \Lambda\Lambda^T + \sigma_g^2 I_n\right]^{-1} \Lambda\,.$$

Conditional distribution of $b_g$ given $y_g$ is, with $p(y_g|b_g)$ as seen in (B.4),

$$p(b_g|y_g) \propto p(y_g|b_g)p(b_g).$$

Hence, $p(b_g|y_g)$ is *Bernoulli*$(\hat{b}_g)$ with

$$\hat{b}_g = \mathbb{P}(b_g = 1|y_g) = \mathbb{E}(b_g|y_g) = \frac{u_g}{u_g + v_g}, \tag{B.5}$$

where $u_g = p \times N_n(y_g;\ \mu + X_c\gamma_g + \mathbf{x}\psi,\ X\Psi_\beta X^T + \Lambda\Lambda^T + \sigma_g^2 I_n)$ and $v_g = (1-p) \times N_n(y_g;\ \mu + X_c\gamma_g,\ XA_0\Psi_\beta X^T + \Lambda\Lambda^T + \sigma_g^2 I_n)$, with $N_n(y_g; \mu_g, \tau_g)$ denoting the multivariate normal density evaluated at $y_g$ with mean $\mu_g$ and covariance $\tau_g$.

Taking expectation of the complete data log likelihood (B.3) with respect to conditional distribution of missing data given $y_g$ is equivalent to taking expectation with respect to $\begin{bmatrix}\beta_g \\ F_g\end{bmatrix} \Big| y_g, b_g$ and then with respect to $b_g|y_g$, for each $g$. This results in

$$Q = \sum_g Q_g$$

where

$$
\begin{aligned}
Q_g &= \mathbb{E}_{b_g}\left\{ \mathbb{E}_{\begin{bmatrix}\beta_g \\ F_g\end{bmatrix}}\left\{ l_g \middle| y_g, b_g \right\} \middle| y_g \right\} \\
&= \mathbb{E}\left\{ \mathbb{E}\left\{ -\frac{n}{2}\log\sigma_g^2 - \frac{1}{2\sigma_g^2}\left\| y_g - \mu - X_c\gamma_g - \mathbf{x}(b_g\psi) - [X,\Lambda]\begin{bmatrix}\beta_g \\ F_g\end{bmatrix}\right\|^2 \right.\right. \\
&\quad -\frac{1}{2}\log\left|((1-b_g)A_0\Psi_\beta + b_g\Psi_\beta)\right| \\
&\quad -\frac{1}{2}\left[\beta_g^T, F_g^T\right]\begin{bmatrix}((1-b_g)A_0\Psi_\beta + b_g\Psi_\beta) & 0 \\ 0 & I_q\end{bmatrix}^{-1}\begin{bmatrix}\beta_g \\ F_g\end{bmatrix} \\
&\quad \left.\left. + b_g\log p + (1-b_g)\log(1-p) \,\middle|\, y_g, b_g \right\} \middle| y_g \right\} \\
&= \mathbb{E}\left\{ -\frac{n}{2}\log\sigma_g^2 - \frac{1}{2}\log\left|((1-b_g)A_0\Psi_\beta + b_g\Psi_\beta)\right| - \frac{1}{2\sigma_g^2}\left\| y_g - \mu - X_c\gamma_g - \mathbf{x}(b_g\psi)\right\|^2 \right. \\
&\quad + \frac{1}{\sigma_g^2}(y_g - \mu - X_c\gamma_g - \mathbf{x}(b_g\psi))^T(X\hat{\beta}_g + \Lambda\hat{F}_g) \\
&\quad -\frac{1}{2}\left(\hat{\beta}_g^T\left(\frac{1}{\sigma_g^2}X^TX + [((1-b_g)A_0\Psi_\beta + b_g\Psi_\beta)]^{-1}\right)\hat{\beta}_g + 2\hat{\beta}_g^T\left(\frac{1}{\sigma_g^2}X^T\Lambda\right)\hat{F}_g \right. \\
&\quad \left. + \hat{F}_g^T\left(\frac{1}{\sigma_g^2}\Lambda^T\Lambda + I_q\right)\hat{F}_g\right) \\
&\quad -\frac{1}{2}tr\left(\left(\frac{1}{\sigma_g^2}X^TX + [((1-b_g)A_0\Psi_\beta + b_g\Psi_\beta)]^{-1}\right)\hat{\Sigma}_{g11}\right) - \frac{1}{2}tr\left(\left(\frac{1}{\sigma_g^2}X^T\Lambda\right)\hat{\Sigma}_{g21}\right) \\
&\quad -\frac{1}{2}tr\left(\left(\frac{1}{\sigma_g^2}\Lambda^TX\right)\hat{\Sigma}_{g12}\right) - \frac{1}{2}tr\left(\left(\frac{1}{\sigma_g^2}\Lambda^T\Lambda + I_q\right)\hat{\Sigma}_{g22}\right) \,\middle|\, y_g \Bigg\} \\
&\quad + \hat{b}_g\log p + (1-\hat{b}_g)\log(1-p)\,.
\end{aligned}
$$

Since $A_0\Psi_\beta = \begin{bmatrix}\sigma_0^2 & 0 \\ 0 & 0\end{bmatrix}$ is singular, we replace its determinant with the product of its positive eigenvalues, and replace its inverse with generalized inverse $(A_0\Psi_\beta)^-$ such that $(A_0\Psi_\beta) = (A_0\Psi_\beta)(A_0\Psi_\beta)^-(A_0\Psi_\beta)$. Hence

$$
\left|A_0\Psi_\beta\right| = \left|\begin{bmatrix}\sigma_0^2 & 0 \\ 0 & 0\end{bmatrix}\right| = \sigma_0^2, \qquad (A_0\Psi_\beta)^- = \begin{bmatrix}\frac{1}{\sigma_0^2} & 0 \\ 0 & 0\end{bmatrix}.
$$

For any function $h(b_g)$ of $b_g$, with $b_g|y_g \sim Bernoulli(\hat{b}_g)$,

$$\mathbb{E}_{b_g}\left(h(b_g)|y_g\right) = \hat{b}_g \times h(1) + (1-\hat{b}_g) \times h(0)\,.$$

This leads to

$$\begin{aligned}\mathbb{E}_{b_g}\left\{\log|((1-b_g)A_0\Psi_\beta + b_g\Psi_\beta)|\Big|\, y_g\right\} &= \hat{b}_g \log|\Psi_\beta| + (1-\hat{b}_g)(\log|A_0\Psi_\beta|) \\ &= \log\sigma_0^2 + \hat{b}_g \log\sigma_1^2\,.\end{aligned}$$

Thus, the E-step estimates are

$$\begin{aligned}
\hat{\beta}_g^{(1)} &:= \hat{\beta}_g\big|_{b_g=1} = \Psi_\beta X^T[X\Psi_\beta X^T + \Lambda\Lambda^T + \sigma_g^2 I_n]^{-1}(y_g - \mu - X_c\gamma_g - \mathbf{x}\psi) \\
\hat{\beta}_g^{(0)} &:= \hat{\beta}_g\big|_{b_g=0} = A_0\Psi_\beta X^T[XA_0\Psi_\beta X^T + \Lambda\Lambda^T + \sigma_g^2 I_n]^{-1}(y_g - \mu - X_c\gamma_g) \\
\hat{F}_g^{(1)} &:= \hat{F}_g\big|_{b_g=1} = \Lambda^T[X\Psi_\beta X^T + \Lambda\Lambda^T + \sigma_g^2 I_n]^{-1}(y_g - \mu - X_c\gamma_g - \mathbf{x}\psi) \\
\hat{F}_g^{(0)} &:= \hat{F}_g\big|_{b_g=0} = \Lambda^T[XA_0\Psi_\beta X^T + \Lambda\Lambda^T + \sigma_g^2 I_n]^{-1}(y_g - \mu - X_c\gamma_g) \\
\hat{\Sigma}_{g11}^{(1)} &:= \hat{\Sigma}_{g11}\big|_{b_g=1} = \Psi_\beta - \Psi_\beta X^T[X\Psi_\beta X^T + \Lambda\Lambda^T + \sigma_g^2 I_n]^{-1}X\Psi_\beta \\
\hat{\Sigma}_{g11}^{(0)} &:= \hat{\Sigma}_{g11}\big|_{b_g=0} = A_0\Psi_\beta - A_0\Psi_\beta X^T[XA_0\Psi_\beta X^T + \Lambda\Lambda^T + \sigma_g^2 I_n]^{-1}XA_0\Psi_\beta \\
\hat{\Sigma}_{g21}^{(1)} &:= \hat{\Sigma}_{g21}\big|_{b_g=1} = -\Lambda^T[X\Psi_\beta X^T + \Lambda\Lambda^T + \sigma_g^2 I_n]^{-1}X\Psi_\beta \\
\hat{\Sigma}_{g21}^{(0)} &:= \hat{\Sigma}_{g21}\big|_{b_g=0} = -\Lambda^T[XA_0\Psi_\beta X^T + \Lambda\Lambda^T + \sigma_g^2 I_n]^{-1}XA_0\Psi_\beta \\
\hat{\Sigma}_{g22}^{(1)} &:= \hat{\Sigma}_{g22}\big|_{b_g=1} = I_q - \Lambda^T[X\Psi_\beta X^T + \Lambda\Lambda^T + \sigma_g^2 I_n]^{-1}\Lambda \\
\hat{\Sigma}_{g22}^{(0)} &:= \hat{\Sigma}_{g22}\big|_{b_g=0} = I_q - \Lambda^T[XA_0\Psi_\beta X^T + \Lambda\Lambda^T + \sigma_g^2 I_n]^{-1}\Lambda\,.
\end{aligned} \tag{B.6}$$

In fact, applying the Woodbury matrix identity

$$\mathbf{AB^T}(\mathbf{BAB^T} + \mathbf{C})^{-1} = (\mathbf{A^{-1}} + \mathbf{B^T C^{-1} B})^{-1}\mathbf{B^T C^{-1}}, \tag{B.7}$$

and denoting $\underset{(2+q)\times(2+q)}{\Psi} = \begin{bmatrix}\Psi_\beta & \mathbf{0} \\ \mathbf{0} & I_q\end{bmatrix}$ and $\underset{n\times(2+q)}{\Omega} = [X, \Lambda]$, we can simplify the E-step

estimates that are conditional on $b_g = 1$ as follows.

$$\hat{\beta}_g^{(1)} = \begin{bmatrix} I_2 & \mathbf{0} \end{bmatrix} \frac{1}{\sigma_g^2} \left[ \Psi^{-1} + \frac{1}{\sigma_g^2} \Omega^T \Omega \right]^{-1} \Omega^T (y_g - \mu - X_c \gamma_g - \mathbf{x}\psi)$$

$$\hat{F}_g^{(1)} = \begin{bmatrix} \mathbf{0} & I_q \end{bmatrix} \frac{1}{\sigma_g^2} \left[ \Psi^{-1} + \frac{1}{\sigma_g^2} \Omega^T \Omega \right]^{-1} \Omega^T (y_g - \mu - X_c \gamma_g - \mathbf{x}\psi)$$

$$\hat{\Sigma}_{g11}^{(1)} = \begin{bmatrix} I_2 & \mathbf{0} \end{bmatrix} \left[ \Psi^{-1} + \frac{1}{\sigma_g^2} \Omega^T \Omega \right]^{-1} \begin{bmatrix} I_2 \\ \mathbf{0} \end{bmatrix}$$

$$\hat{\Sigma}_{g21}^{(1)} = \begin{bmatrix} \mathbf{0} & I_q \end{bmatrix} \left[ \Psi^{-1} + \frac{1}{\sigma_g^2} \Omega^T \Omega \right]^{-1} \begin{bmatrix} I_2 \\ \mathbf{0} \end{bmatrix}$$

$$\hat{\Sigma}_{g22}^{(1)} = \begin{bmatrix} \mathbf{0} & I_q \end{bmatrix} \left[ \Psi^{-1} + \frac{1}{\sigma_g^2} \Omega^T \Omega \right]^{-1} \begin{bmatrix} \mathbf{0} \\ I_q \end{bmatrix}.$$

The complete $Q$ function is $Q = \sum_g Q_g$ where

$$
\begin{aligned}
Q_g &= \hat{b}_g \times Q_g\big|_{b_g=1} + (1-\hat{b}_g) \times Q_g\big|_{b_g=0} \\
&= -\frac{n}{2}\log\sigma_g^2 - \frac{1}{2}\left(\log\sigma_0^2 + \hat{b}_g \log\sigma_1^2\right) \\
&\quad + \hat{b}_g \times \left\{ -\frac{1}{2\sigma_g^2}\left\|y_g - \mu - X_c\gamma_g - \mathbf{x}\psi\right\|^2 \right. \\
&\qquad + \frac{1}{\sigma_g^2}\left(y_g - \mu - X_c\gamma_g - \mathbf{x}\psi\right)^T \left(X\hat{\beta}_g^{(1)} + \Lambda\hat{F}_g^{(1)}\right) \\
&\qquad - \frac{1}{2}\hat{\beta}_g^{(1)T}\left(\frac{1}{\sigma_g^2}X^TX + \begin{bmatrix} \frac{1}{\sigma_0^2} & 0 \\ 0 & \frac{1}{\sigma_1^2} \end{bmatrix}\right)\hat{\beta}_g^{(1)} - \hat{\beta}_g^{(1)T}\left(\frac{1}{\sigma_g^2}X^T\Lambda\right)\hat{F}_g^{(1)} \\
&\qquad - \frac{1}{2}\hat{F}_g^{(1)T}\left(\frac{1}{\sigma_g^2}\Lambda^T\Lambda + I_q\right)\hat{F}_g^{(1)} \\
&\qquad - \frac{1}{2}tr\left(\left(\frac{1}{\sigma_g^2}X^TX + \begin{bmatrix} \frac{1}{\sigma_0^2} & 0 \\ 0 & \frac{1}{\sigma_1^2} \end{bmatrix}\right)\hat{\Sigma}_{g11}^{(1)}\right) - tr\left(\left(\frac{1}{\sigma_g^2}X^T\Lambda\right)\hat{\Sigma}_{g21}^{(1)}\right) \\
&\qquad \left. - \frac{1}{2}tr\left(\left(\frac{1}{\sigma_g^2}\Lambda^T\Lambda + I_q\right)\hat{\Sigma}_{g22}^{(1)}\right) + \log p \right\} \\
&\quad + (1-\hat{b}_g) \times \left\{ -\frac{1}{2\sigma_g^2}\left\|y_g - \mu - X_c\gamma_g\right\|^2 \right. \\
&\qquad + \frac{1}{\sigma_g^2}\left(y_g - \mu - X_c\gamma_g\right)^T \left(X\hat{\beta}_g^{(0)} + \Lambda\hat{F}_g^{(0)}\right) \\
&\qquad - \frac{1}{2}\hat{\beta}_g^{(0)T}\left(\frac{1}{\sigma_g^2}X^TX + \begin{bmatrix} \frac{1}{\sigma_0^2} & 0 \\ 0 & 0 \end{bmatrix}\right)\hat{\beta}_g^{(0)} - \hat{\beta}_g^{(0)T}\left(\frac{1}{\sigma_g^2}X^T\Lambda\right)\hat{F}_g^{(0)} \\
&\qquad - \frac{1}{2}\hat{F}_g^{(0)T}\left(\frac{1}{\sigma_g^2}\Lambda^T\Lambda + I_q\right)\hat{F}_g^{(0)} \\
&\qquad - \frac{1}{2}tr\left(\left(\frac{1}{\sigma_g^2}X^TX + \begin{bmatrix} \frac{1}{\sigma_0^2} & 0 \\ 0 & 0 \end{bmatrix}\right)\hat{\Sigma}_{g11}^{(0)}\right) - tr\left(\left(\frac{1}{\sigma_g^2}X^T\Lambda\right)\hat{\Sigma}_{g21}^{(0)}\right) \\
&\qquad \left. - \frac{1}{2}tr\left(\left(\frac{1}{\sigma_g^2}\Lambda^T\Lambda + I_q\right)\hat{\Sigma}_{g22}^{(0)}\right) + \log(1-p) \right\} .
\end{aligned}
\tag{B.8}
$$

### B.1.2 M-step

For each parameter, with the other parameters and E-step estimates held fixed at their current values, conditional maximization of $Q$ with respect to the parameter is conducted.

Maximizing $Q = \sum_g Q_g$ with respect to $\mu$ yields

$$\hat{\mu} = \frac{1}{\sum_g \left(\frac{1}{\sigma_g^2}\right)} \sum_g \frac{1}{\sigma_g^2} \left(y_g - X_c\gamma_g - \hat{b}_g\left(\mathbf{x}\psi + X\hat{\beta}_g^{(1)} + \Lambda\hat{F}_g^{(1)}\right) - \left(1-\hat{b}_g\right)\left(X\hat{\beta}_g^{(0)} + \Lambda\hat{F}_g^{(0)}\right)\right).$$

Maximizing $Q = \sum_g Q_g$ with respect to $\sigma_0^2$ yields

$$\hat{\sigma}_0^2 = \frac{1}{G}\sum_g \left(\hat{b}_g\left(\left[\hat{\beta}_g^{(1)}\right]_1^2 + \left[\hat{\Sigma}_{g11}^{(1)}\right]_{11}\right) + \left(1-\hat{b}_g\right)\left(\left[\hat{\beta}_g^{(0)}\right]_1^2 + \left[\hat{\Sigma}_{g11}^{(0)}\right]_{11}\right)\right),$$

where $\left[\hat{\beta}_g^{(1)}\right]_1$ stands for the first element of the vector $\hat{\beta}_g^{(1)}$, $\left[\hat{\beta}_g^{(1)}\right]_1^2$ its square, and $\left[\hat{\Sigma}_{g11}^{(1)}\right]_{11}$ the $(1,1)$ element of the $2\times 2$ matrix $\hat{\Sigma}_{g11}^{(1)}$, and so on.

Maximizing $Q = \sum_g Q_g$ with respect to $\sigma_1^2$ yields

$$\hat{\sigma}_1^2 = \frac{1}{\left(\sum_g \hat{b}_g\right)}\sum_g \left(\hat{b}_g\left(\left[\hat{\beta}_g^{(1)}\right]_2^2 + \left[\hat{\Sigma}_{g11}^{(1)}\right]_{22}\right)\right).$$

Maximizing $Q = \sum_g Q_g$ with respect to $\psi$ yields

$$\hat{\psi} = \frac{1}{\sum_g \left(\frac{\hat{b}_g}{\sigma_g^2}\mathbf{x}^T\mathbf{x}\right)} \sum_g \left\{\frac{\hat{b}_g}{\sigma_g^2}\left(y_g - \mu - \left(X\hat{\beta}_g^{(1)} + \Lambda\hat{F}_g^{(1)}\right)\right)^T \mathbf{x}\right\}.$$

Maximizing $Q = \sum_g Q_g$ with respect to $\gamma_g$ for each $g$ yields

$$\hat{\gamma}_g = \left(X_c^T X_c\right)^{-1} X_c^T \left(y_g - \mu - \hat{b}_g\left(\mathbf{x}\psi + X\hat{\beta}_g^{(1)} + \Lambda\hat{F}_g^{(1)}\right) - \left(1-\hat{b}_g\right)\left(X\hat{\beta}_g^{(0)} + \Lambda\hat{F}_g^{(0)}\right)\right).$$

Maximizing $Q = \sum_g Q_g$ with respect to $\Lambda$ yields

$$\hat{\Lambda} = \left[\sum_g \frac{1}{\sigma_g^2}\left(\hat{b}_g\left(\left(y_g - \mu - X_c\gamma_g - \mathbf{x}\psi - X\hat{\beta}_g^{(1)}\right)\hat{F}_g^{(1)T} - \frac{1}{2}X\left(\hat{\Sigma}_{g21}^{(1)\,T} + \hat{\Sigma}_{g12}^{(1)}\right)\right)\right.\right.$$
$$\left.\left. + \left(1-\hat{b}_g\right)\left(\left(y_g - \mu - X_c\gamma_g - X\hat{\beta}_g^{(0)}\right)\hat{F}_g^{(0)T} - \frac{1}{2}X\left(\hat{\Sigma}_{g21}^{(0)\,T} + \hat{\Sigma}_{g12}^{(0)}\right)\right)\right)\right]$$
$$\left[\sum_g \frac{1}{\sigma_g^2}\left(\hat{b}_g\left(\hat{F}_g^{(1)}\hat{F}_g^{(1)T} + \hat{\Sigma}_{g22}^{(1)}\right) + \left(1-\hat{b}_g\right)\left(\hat{F}_g^{(0)}\hat{F}_g^{(0)T} + \hat{\Sigma}_{g22}^{(0)}\right)\right)\right]^{-1}.$$

Maximizing $Q = \sum_g Q_g$ with respect to $p$ yields

$$\hat{p} = \frac{1}{G} \sum_g \hat{b}_g \, .$$

### B.1.3 Shrinkage estimation of gene-specific error variance

We adopt the method in Bar et al. (2010) for producing shrinkage estimate of $\{\sigma_g^2\}$. We first obtain gene-specific mean squared error $\{m_g\}$ from the linear model of $y_g$ on five fixed covariates: SNP information $\mathbf{x}$, gender, disease status, smoking status, and loading of first factor from factor analysis on genotype (SNP) covariance matrix. That is, no hidden effect term is involved in this simple model with only fixed effects. With $X_a = [X, X_c]$ where $X$ and $X_c$ are as defined in model (B.1) and $\eta_g = \begin{bmatrix} \beta_g \\ \gamma_g \end{bmatrix}$, the simple linear regression model is

$$\begin{aligned} y_g &= X_a \eta_g + W_g \\ W_g &\sim N_n(\mathbf{0}, \sigma_g^2 I_n). \end{aligned} \tag{B.9}$$

Residuals of such model fitted for a given gene $g$ give rise to the gene-specific mean squared error $m_g$ for that gene.

Since there is no hidden effect involved in the simple linear regression model (B.9), $\{m_g\}$ calculated from observed data approximately follows a chi-squared distribution

$$m_g | \sigma_g^2 \sim \sigma_g^2 \frac{\chi_{f_g}^2}{f_g} \tag{B.10}$$

where $f_g = n - rank(X_a)$ for all $g$. Although the chi-squared distribution may not reflect the true distribution of $\{m_g\}$ exactly due to the hidden effect presented in the data, (B.10) suffices to give rise to a shrinkage estimate of $\{\sigma_g^2\}$ via maximum likelihood as noted in Bar et al. (2010). That is, maximum likelihood estimates

of hyperparameters $A, B$ of prior distribution $\sigma_g^2 \sim Inverse\text{-}Gamma(A, B)$ are obtained by maximizing the marginal likelihood of $\{m_g\}$ via the `nlminb` function in R. Thus, posterior mode of $\sigma_g^2$ given $m_g$ can be used as shrinkage estimate of $\sigma_g^2$. That is, for each gene $g$,

$$\hat{\sigma}_g^2 = \left(\frac{f_g/2}{f_g/2 + \hat{A} + 1}\right) m_g + \left(\frac{\hat{A} + 1}{f_g/2 + \hat{A} + 1}\right)\left(\frac{\hat{B}}{\hat{A} + 1}\right). \tag{B.11}$$

### B.1.4 Iterative Scheme

The algorithm is implemented such that with an initial fit of model (B.9) which leads to shrinkage estimate of $\{\sigma_g^2\}$, a first cycle of E- and M-steps above with $\{\sigma_g^2\}$ held fixed at the current estimate is carried out. Prior to performing the second cycle of E- and M-steps, $m_g$ for each $g$ is updated by re-fitting model (B.9) but with an "adjusted" $y_g$ instead that takes into account current values of $\Lambda$ and $F_g$, i.e.

$$y_{g_{adj}} = y_g - \hat{\Lambda}\hat{F}_g, \tag{B.12}$$

and shrinkage estimate of $\{\sigma_g^2\}$ is in turn updated. The second cycle of E- and M-steps is then carried out with $\{\sigma_g^2\}$ held fixed at the updated value. Subsequently, in each iteration until convergence is observed in $Q$, re-fitting the linear model (B.9) with updated $\{y_{g_{adj}}\}$ to update $\{\sigma_g^2\}$ precedes a cycle of E- and M-steps. The idea behind this adaptive scheme is that, for any gene $g$, as the hidden effect $\Lambda F_g$ is updated in each iteration, $y_{g_{adj}}$ would become better proxy for the hidden-variation-removed data, and shrinkage estimate (B.11) resulted from fitting model (B.9) with $y_{g_{adj}}$ would become closer to the true value of $\sigma_g^2$.

To summarize, the main steps of the algorithm are

i). Initialize $\mu, \sigma_0^2, \sigma_1^2, \psi, \{\gamma_g\}, \Lambda, p$.

ii). Obtain $\{m_g\}$ from linear model (B.9) so that $\{\sigma_g^2\}$ is initialized to its shrinkage estimate (B.11).

iii). Use (B.5) and (B.6) to update imputed missing data conditional on current estimate of $\{\sigma_g^2\}$.

iv). Use M-step formulas to update parameters conditional on current estimate of $\{\sigma_g^2\}$.

v). Perform ii) but with "adjusted" response (B.12) to update $\{\sigma_g^2\}$. Then repeat iii) and iv).

vi). Repeat v) until the increase in $Q = \sum_g Q_g$ (B.8) is negligible.

**Initial values**

We use the following as initial values for $\mu, \sigma_0^2, \sigma_1^2, p$ and $\{\gamma_g\}$:

$$\underset{n\times 1}{\mu} = \frac{1}{G}\sum_g y_g, \qquad \sigma_0^2 = 1, \qquad \sigma_1^2 = 0.1,$$

$$p = 0.0005, \qquad \underset{4\times 1}{\gamma_g} = \mathbf{0} \quad \text{for each } g.$$

**Choice of prior variances**

As $\beta_{0g}$ represents a gene-specific offset, we have little prior knowledge on its distribution. Therefore in order to have the factor analysis term capture unwanted variation without interference, using a standard normal as the prior for $\beta_{0g}$ appears sensible. On the other hand, as we envision that $\beta_{1g}$ for genes that are significantly associated with the given SNP are likely to cluster with a mean away from 0, giving $\beta_{1g}$ a relatively small prior variance and allowing its prior mean to be data-driven is a natural thing to do. Another plausible choice is to

assign a large prior variance (and a close to zero prior mean) to $\beta_{1g}$ in an attempt to reflect possible values of $\beta_{1g}$ for non-null genes in the case where $\beta_{1g}$ for non-null genes takes the most extreme (i.e. most positive or most negative) values among all $\beta_{1g}$, $g = 1, ..., G$. However, in practice we found that such choice for the prior distribution of $\beta_{1g}$ may not target the true non-null region enough to classify majority of the non-null genes correctly.

**Initializing $\psi$**

Initial value of $\psi$ depends on the initial value of $p$ which is determined based on the presumption that significant associations are sparse among all associations. After fitting the linear model (B.9), we obtain estimate of the SNP effect $\{\delta_g\}$ on each gene, sort all the genes by $\{\delta_g\}$, and determine the initial value of $\psi$ based on the following scheme. Let the mean of the $p \times G$ smallest $\delta_g$ values be $L_1$ and the mean of the remaining $(1-p)\times G$ values of $\delta_g$ be $L_0$. Similarly, let the mean of the $p \times G$ largest $\delta_g$ values be $R_1$ and the mean of the remaining $(1-p)\times G$ values of $\delta_g$ be $R_0$. If $|L_1 - L_0| > |R_1 - R_0|$ then set $\psi = L_1 - L_0$, otherwise set $\psi = R_1 - R_0$. This initial value of $\psi$ reflects possible difference in location between center of non-null genes and center of null genes, where null or non-null is with respect to the given SNP.

**Initializing $\Lambda$**

Initial value of $\Lambda$ is obtained via spectral decomposition. Let $X_a = [X, X_c]$. With hat matrix $H = X_a(X_a^T X_a)^{-1} X_a^T$ and $S_g = (I_n - H) y_g y_g^T (I_n - H)$, $S = \frac{1}{G}\sum_g S_g$ is an

estimate of the covariance matrix of $Y$, i.e.

$$S \approx \Lambda\Lambda^T + \hat{\sigma}^2 I_n$$

where $\hat{\sigma}^2 = \frac{1}{G}\sum_g \hat{\sigma}_g^2$. Let $\Delta_q$ be the $n \times q$ matrix of eigenvectors of $S - \hat{\sigma}^2 I_n$ that correspond to the $q$ largest eigenvalues $a_1, ..., a_q$, and

$$A_q = \text{diag}\left(\sqrt{a_k},\ k = 1, \ldots, q\right).$$

Then we set $\Lambda = \Delta_q A_q$ as initial value of the hidden factor loading matrix $\Lambda$.

## B.2 FF model and EM algorithm

For any gene $g$, let $\underset{n\times 6}{X_a} = [X, X_c]$ and $\underset{6\times 1}{\eta_g} = \begin{bmatrix}\beta_g \\ \gamma_g\end{bmatrix}$, where $X$ and $X_c$ are as defined in model (B.1), the model is a simplification of the RRmix model so that both the SNP effect $\beta_g$ and the error variance are treated as fixed effects and the mixture structure is removed.

$$y_g = \mu + X_a\eta_g + \Lambda F_g + W_g$$

$$W_g \sim N_n(0, \sigma_g^2 I_n)$$

$$F_g \sim N_q(0, I_q).$$

Parameters to be estimated: $\Theta = \left\{\mu, \Lambda, \{\eta_g\}, \{\sigma_g^2\}\right\}$.

Missing data: $\{F_g\}$.

The complete data log-likelihood is $l = \sum_g l_g$ where $l_g$ is the complete data log likelihood for gene $g$, i.e.

$$\begin{aligned} l_g &= \log p(y_g|F_g;\Theta) + \log p(F_g) \\ &= -\frac{n}{2}\log\sigma_g^2 - \frac{1}{2\sigma_g^2}\left\|y_g - \mu - X_a\eta_g - \Lambda F_g\right\|^2 - \frac{1}{2}\left\|F_g\right\|^2. \end{aligned}$$

### B.2.1 E-step

The model implies the following $(q+n)$-dimensional multivariate normal distribution

$$\begin{bmatrix} F_g \\ y_g \end{bmatrix} \sim N\left( \begin{bmatrix} 0_q \\ \mu + X_a\eta_g \end{bmatrix}, \quad \begin{bmatrix} I_q & \Lambda^T \\ \Lambda & \Lambda\Lambda^T + \sigma_g^2 I_n \end{bmatrix} \right)$$

which then implies

$$\begin{aligned} \hat{F}_g &= \mathbb{E}\left(F_g|y_g\right) = \Lambda^T[\Lambda\Lambda^T + \sigma_g^2 I_n]^{-1}(y_g - \mu - X_a\eta_g) \\ &= \frac{1}{\sigma_g^2}\left[I_q + \frac{1}{\sigma_g^2}\Lambda^T\Lambda\right]^{-1}\Lambda^T(y_g - \mu - X_a\eta_g) \\ \hat{\Sigma}_g &= \mathbb{V}\left(F_g|y_g\right) = I_q - \Lambda^T[\Lambda\Lambda^T + \sigma_g^2 I_n]^{-1}\Lambda \\ &= \left[I_q + \frac{1}{\sigma_g^2}\Lambda^T\Lambda\right]^{-1}. \end{aligned} \tag{B.13}$$

where simplified formulas result from application of the Woodbury identity.

Taking expectation of the complete data log likelihood with respect to conditional distribution of missing data given $y_g$ results in $Q = \sum_g Q_g$ where

$$\begin{aligned} Q_g = -\frac{n}{2}\log\sigma_g^2 - \frac{1}{2\sigma_g^2}\left\|y_g - \mu - X_a\eta_g\right\|^2 + \frac{1}{\sigma_g^2}\left(y_g - \mu - X_a\eta_g\right)^T\Lambda\hat{F}_g \\ - \frac{1}{2}\hat{F}_g^T\left(\frac{1}{\sigma_g^2}\Lambda^T\Lambda + I_q\right)\hat{F}_g - \frac{1}{2}tr\left(\left(\frac{1}{\sigma_g^2}\Lambda^T\Lambda + I_q\right)\hat{\Sigma}_g\right). \end{aligned} \tag{B.14}$$

### B.2.2 M-step

For each parameter, the ML estimate is obtained via conditional maximization of $Q$ with respect to the parameter with the other parameters held fixed at their

current values.

$$
\begin{aligned}
\hat{\mu} &= \frac{1}{\sum_g 1/\sigma_g^2} \sum_g \frac{1}{\sigma_g^2} \left( y_g - X_a \eta_g - \Lambda \hat{F}_g \right) \\
\hat{\Lambda} &= \left[ \sum_g \frac{1}{\sigma_g^2} \left( y_g - \mu - X_a \eta_g \right) \hat{F}_g^T \right] \left[ \sum_g \frac{1}{\sigma_g^2} \left( \hat{F}_g \hat{F}_g^T + \hat{\Sigma}_g \right) \right]^{-1} \\
\hat{\eta}_g &= \left( X_a^T X_a \right)^{-1} X_a^T \left( y_g - \mu - \Lambda \hat{F}_g \right) \\
\hat{\sigma}_g^2 &= \frac{1}{n} \left( \left\| y_g - \mu - X_a \eta_g - \Lambda F_g \right\|^2 + tr \left( \Lambda^T \Lambda \hat{\Sigma}_g \right) \right)
\end{aligned}
\tag{B.15}
$$

### B.2.3 Iterative Scheme

To summarize, the main steps of the algorithm are

i). Initialize $\Theta = \left\{ \mu, \Lambda, \{\eta_g\}, \{\sigma_g^2\} \right\}$.

ii). Update missing data in E-step (B.13) conditional on current values of $\Theta$.

iii). Update $\Theta$ in M-step (B.15) conditional on current value of missing data.

iv). Iterate cycles consisting of ii) and iii) until the increase in $Q = \sum_g Q_g$ (B.14) is negligible.

#### Initial values

We use the same initial value for $\mu$ and $\Lambda$ as in RRmix. Also, we use initialize $\{\sigma_g^2\}$ to its initial shrinkage estimate of in RRmix and $\{\eta_g\}$ to a zero matrix.

### B.2.4 Calculation of p-values

The expected information matrix $I(\hat{\eta}_g)$ is estimated by the observed information matrix $-\frac{\partial^2 l_{obs}(Y;\Theta)}{\partial \eta_g \partial \eta_g^T}$ where $l_{obs}(Y;\Theta)$ is the observed data log-likelihood. Based

on equation (6) in Oakes (1999), a formula for calculating $\frac{\partial^2 l_{obs}(Y;\Theta)}{\partial\eta_g\partial\eta_g^T}$ is

$$\frac{\partial^2 l_{obs}(Y;\Theta)}{\partial\eta_g\partial\eta_g^T} = \frac{\partial^2 Q(\eta_g|\hat{\eta}_g)}{\partial\eta_g\partial\eta_g^T}\bigg|_{\eta_g=\hat{\eta}_g} + \frac{\partial^2 Q(\eta_g|\hat{\eta}_g)}{\partial\eta_g\partial\hat{\eta}_g^T}\bigg|_{\eta_g=\hat{\eta}_g}.$$

Here, $Q(\eta_g|\hat{\eta}_g)$ is the $Q$ function written explicitly where the E-step estimates are replaced with their update formulas that are functions of $\eta_g$, so that $Q$ is a function of $\eta_g$ conditional on $\hat{\eta}_g$ and it can be interpreted that given the parameter estimates from the $k$-th iteration, the estimate of $\eta_g$ for the $(k+1)$-th iteration is obtained as $\hat{\eta}_g^{(k+1)} = \text{argmax } Q\left(\eta_g\middle|\hat{\eta}_g^{(k)}\right)$.

Since $Q = \sum_g Q_g$, we have

$$\begin{aligned}
\frac{\partial^2 Q(\eta_g|\hat{\eta}_g)}{\partial\eta_g\partial\eta_g^T}\bigg|_{\eta_g=\hat{\eta}_g} &= -\frac{1}{\sigma_g^2}X_a^T X_a \\
\frac{\partial^2 Q(\eta_g|\hat{\eta}_g)}{\partial\eta_g\partial\hat{\eta}_g^T}\bigg|_{\eta_g=\hat{\eta}_g} &= \left\{\frac{\partial}{\partial\hat{\eta}_g^T}\frac{1}{\sigma_g^2}X_a^T\left(y_g-\mu-X_a\eta_g-\Lambda\hat{F}_g\right)\right\}\bigg|_{\eta_g=\hat{\eta}_g} \\
&= \left\{\frac{\partial}{\partial\hat{\eta}_g^T}\frac{1}{\sigma_g^2}X_a^T\left(y_g-\mu-X_a\eta_g-\Lambda\Lambda^T\left[\Lambda\Lambda^T+\sigma_g^2 I_n\right]^{-1}(y_g-\mu-X_a\hat{\eta}_g)\right)\right\}\bigg|_{\eta_g=\hat{\eta}_g} \\
&= \frac{1}{\sigma_g^2}X_a^T\Lambda\Lambda^T\left[\Lambda\Lambda^T+\sigma_g^2 I_n\right]^{-1}X_a\,.
\end{aligned}$$

Then, with each parameter at its estimated value upon convergence of the EM algorithm,

$$\begin{aligned}
I(\hat{\eta}_g) &\approx -\left(-\frac{1}{\sigma_g^2}X_a^T X_a + \frac{1}{\sigma_g^2}X_a^T\Lambda\Lambda^T\left[\Lambda\Lambda^T+\sigma_g^2 I_n\right]^{-1}X_a\right) \\
&= \frac{1}{\sigma_g^2}X_a^T\left(I_n-\Lambda\Lambda^T\left[\Lambda\Lambda^T+\sigma_g^2 I_n\right]^{-1}\right)X_a \\
&= \frac{1}{\sigma_g^2}X_a^T\left(I_n-\frac{1}{\sigma_g^2}\Lambda\left[I_q+\frac{1}{\sigma_g^2}\Lambda^T\Lambda\right]^{-1}\Lambda^T\right)X_a.
\end{aligned}$$

The t-statistic for the gene-SNP association $\beta_{1g}$ is obtained from taking the corresponding elements in $\hat{\eta}_g$ and $I(\hat{\eta}_g)^{-1}$, and is

$$\frac{[\hat{\eta}_g]_2}{\sqrt{[I(\hat{\eta}_g)^{-1}]_{2,2}}}.$$

Under t-distribution with degrees of freedom $n-(1+rank(X_a)+rank(\Lambda))$, or the standard normal distribution as the degrees of freedom is large enough for the

t-distribution to be well approximated by the standard normal distribution, p-value for conducting the hypothesis test with $H_0 : \hat{\beta}_{1g} = 0$, i.e. no association between gene $g$ and the given SNP, is calculated. Bonferroni correction on the significance level 0.05 is employed, so that each p-value is compared against $\frac{0.05}{G \times N}$, where $G$ is the total number of genes and $N$ is the total number of SNPs.

## B.3 FR model and EM algorithm

For any gene $g$, the model is the same as the FF model, except that the error variance is assigned a prior. That is,

$$y_g = \mu + X_a \eta_g + \Lambda F_g + W_g$$

$$W_g \sim N_n(0, \sigma_g^2 I_n)$$

$$F_g \sim N_q(0, I_q)$$

$$\sigma_g^2 \sim IG(A, B)$$

Parameters to be estimated: $\Theta = \left\{\mu, \Lambda, \{\eta_g\}\right\}$.

Shrinkage estimate to be calculated: $\{\sigma_g^2\}$.

Missing data: $\{F_g\}$.

### Similarity to the FF model

The estimation formulas in E- and M-steps and the $Q$ function are the same as in the FF model. However, error variance $\{\sigma_g^2\}$ is no longer estimated in the M-step. Shrinkage estimation of $\{\sigma_g^2\}$ is performed the same way as described in the algorithm for the RRmix model.

## Iterative Scheme

To summarize, the main steps of the algorithm are

i). Initialize $\Theta = \left\{\mu, \Lambda, \{\eta_g\}\right\}$.

ii). Initialize shrinkage estimate of $\{\sigma_g^2\}$ (B.11).

iii). Update missing data in E-step (B.13) conditional on current value of $\Theta$ and shrinkage estimate of $\{\sigma_g^2\}$.

iv). Update $\Theta$ in M-step (B.15) conditional on current value of missing data and shrinkage estimate of $\{\sigma_g^2\}$.

v). Iterate cycles consisting of ii), iii) and iv) until the increase in $Q = \sum_g Q_g$ (B.14) is negligible.

### Initial values

We use the same initial value for $\mu$ and $\Lambda$ as in RRmix. Also, we use initialize $\{\sigma_g^2\}$ to its initial shrinkage estimate in RRmix and $\{\eta_g\}$ to a zero matrix.

### Calculation of p-values

p-values are calculated in the same way as in the FF model, under the standard normal distribution. The difference in p-values and in turn in the outcome of the multiple hypothesis tests between the FF and FR models lies in the difference in value between ML estimate of the error variance $\{\sigma_g^2\}$ which is treated as a fixed effect in FF and shrinkage estimate of $\{\sigma_g^2\}$ which is treated as a random effect in FR.

## B.4 RF model and EM algorithm

The RF model specification is the similar to the FF model, but a prior is assigned for primary effect $\beta_g$ which is treated as random.

$$
\begin{aligned}
y_g &= \mu + X_c\gamma_g + X\beta_g + \Lambda F_g + W_g \\
W_g &\sim N_n(0, \sigma_g^2 I_n) \\
F_g &\sim N_q(0, I_q) \\
\beta_g &\sim N_2\left( 0\,, \begin{bmatrix} \sigma_0^2 & 0 \\ 0 & \sigma_1^2 \end{bmatrix} \right).
\end{aligned}
$$

Parameters to be estimated: $\Theta = \left\{\mu, \Lambda, \{\gamma_g\}, \sigma_0^2, \sigma_1^2, \{\sigma_g^2\}\right\}$.
Missing data: $\left\{\{\beta_g\}, \{F_g\}\right\}$.

The complete data log-likelihood is $l = \sum_g l_g$ where $l_g$ is the complete data log likelihood for gene $g$

$$
\begin{aligned}
l_g &= \log p(y_g|\beta_g, F_g; \Theta) + \log p(F_g) + \log p(\beta_g; \sigma_0^2, \sigma_1^2) \\
&= -\frac{n}{2}\log\sigma_g^2 - \frac{1}{2\sigma_g^2}\left\|y_g - \mu - X_c\gamma_g - X\beta_g - \Lambda F_g\right\|^2 - \frac{1}{2}\left\|F_g\right\|^2 \\
&\quad - \frac{1}{2}\log(\sigma_0^2\sigma_1^2) - \frac{1}{2}\beta_g^T \begin{bmatrix} \frac{1}{\sigma_0^2} & 0 \\ 0 & \frac{1}{\sigma_1^2} \end{bmatrix} \beta_g\,.
\end{aligned}
$$

## E-step

With $\Psi_\beta = \begin{bmatrix} \sigma_0^2 & 0 \\ 0 & \sigma_1^2 \end{bmatrix}$, the model implies the following $(2 + q + n)$-dimensional multivariate normal distribution

$$
\begin{bmatrix} \begin{bmatrix} \beta_g \\ F_g \end{bmatrix} \\ y_g \end{bmatrix} \sim N\left( \begin{bmatrix} 0_{2+q} \\ \mu + X_c\gamma_g \end{bmatrix}, \begin{bmatrix} \begin{bmatrix} \Psi_\beta & \\ & I_q \end{bmatrix} & \begin{bmatrix} \Psi_\beta X^T \\ \Lambda^T \end{bmatrix} \\ \left[X\Psi_\beta, \Lambda\right] & X\Psi_\beta X^T + \Lambda\Lambda^T + \sigma_g^2 I_n \end{bmatrix} \right).
$$

which then implies,

$$\begin{aligned}
\hat{\beta}_g &= \mathbb{E}\left(\beta_g | y_g\right) = \Psi_\beta X^T [X\Psi_\beta X^T + \Lambda\Lambda^T + \sigma_g^2 I_n]^{-1}(y_g - \mu - X_c\gamma_g) \\
\hat{F}_g &= \mathbb{E}\left(F_g | y_g\right) = \Lambda^T [X\Psi_\beta X^T + \Lambda\Lambda^T + \sigma_g^2 I_n]^{-1}(y_g - \mu - X_c\gamma_g) \\
\hat{\Sigma}_{g11} &= \Psi_\beta - \Psi_\beta X^T [X\Psi_\beta X^T + \Lambda\Lambda^T + \sigma_g^2 I_n]^{-1} X\Psi_\beta \\
\hat{\Sigma}_{g21} &= -\Lambda^T [X\Psi_\beta X^T + \Lambda\Lambda^T + \sigma_g^2 I_n]^{-1} X\Psi_\beta \\
\hat{\Sigma}_{g22} &= I_q - \Lambda^T [X\Psi_\beta X^T + \Lambda\Lambda^T + \sigma_g^2 I_n]^{-1}\Lambda
\end{aligned}$$

where $\hat{\Sigma}_{g11}$, $\hat{\Sigma}_{g21}$, and $\hat{\Sigma}_{g22}$ are block matrices in $\hat{\Sigma}_g = \mathbb{V}\left(\begin{bmatrix}\beta_g \\ F_g\end{bmatrix} \middle| y_g\right) = \begin{bmatrix}\hat{\Sigma}_{g11} & \hat{\Sigma}_{g12} \\ \hat{\Sigma}_{g21} & \hat{\Sigma}_{g22}\end{bmatrix}$.

Applying the Woodbury identity with $\underset{(2+q)\times(2+q)}{\Psi} = \begin{bmatrix}\Psi_\beta & \mathbf{0} \\ \mathbf{0} & I_q\end{bmatrix}$ and $\underset{n\times(2+q)}{\Omega} = [X, \Lambda]$ gives

$$\begin{aligned}
\hat{\beta}_g &= \begin{bmatrix} I_2 & \mathbf{0}\end{bmatrix} \frac{1}{\sigma_g^2}\left[\Psi^{-1} + \frac{1}{\sigma_g^2}\Omega^T\Omega\right]^{-1} \Omega^T\left(y_g - \mu - X_c\gamma_g\right) \\
\hat{F}_g &= \begin{bmatrix} \mathbf{0} & I_q\end{bmatrix} \frac{1}{\sigma_g^2}\left[\Psi^{-1} + \frac{1}{\sigma_g^2}\Omega^T\Omega\right]^{-1} \Omega^T\left(y_g - \mu - X_c\gamma_g\right) \\
\hat{\Sigma}_{g11} &= \begin{bmatrix} I_2 & \mathbf{0}\end{bmatrix}\left[\Psi^{-1} + \frac{1}{\sigma_g^2}\Omega^T\Omega\right]^{-1}\begin{bmatrix} I_2 \\ \mathbf{0}\end{bmatrix} \\
\hat{\Sigma}_{g21} &= \begin{bmatrix} \mathbf{0} & I_q\end{bmatrix}\left[\Psi^{-1} + \frac{1}{\sigma_g^2}\Omega^T\Omega\right]^{-1}\begin{bmatrix} I_2 \\ \mathbf{0}\end{bmatrix} \\
\hat{\Sigma}_{g22} &= \begin{bmatrix} \mathbf{0} & I_q\end{bmatrix}\left[\Psi^{-1} + \frac{1}{\sigma_g^2}\Omega^T\Omega\right]^{-1}\begin{bmatrix} \mathbf{0} \\ I_q\end{bmatrix}.
\end{aligned} \tag{B.16}$$

Taking expectation of the complete data log likelihood with respect to conditional distribution of missing data given $y_g$ results in $Q = \sum_g Q_g$ where

$$\begin{aligned} Q_g = & -\frac{n}{2}\log\sigma_g^2 - \frac{1}{2}\log(\sigma_0^2\sigma_1^2) - \frac{1}{2\sigma_g^2}\left\|y_g - \mu - X_c\gamma_g\right\|^2 \\ & + \frac{1}{\sigma_g^2}\left(y_g - \mu - X_c\gamma_g\right)^T\left(X\hat{\beta}_g + \Lambda\hat{F}_g\right) \\ & - \frac{1}{2}\hat{\beta}_g^T\left(\frac{1}{\sigma_g^2}X^TX + \begin{bmatrix}\frac{1}{\sigma_0^2} & 0\\ 0 & \frac{1}{\sigma_1^2}\end{bmatrix}\right)\hat{\beta}_g - \hat{\beta}_g^T\left(\frac{1}{\sigma_g^2}X^T\Lambda\right)\hat{F}_g \\ & - \frac{1}{2}\hat{F}_g^T\left(\frac{1}{\sigma_g^2}\Lambda^T\Lambda + I_q\right)\hat{F}_g \\ & - \frac{1}{2}tr\left(\left(\frac{1}{\sigma_g^2}X^TX + \begin{bmatrix}\frac{1}{\sigma_0^2} & 0\\ 0 & \frac{1}{\sigma_1^2}\end{bmatrix}\right)\hat{\Sigma}_{g11}\right) - tr\left(\left(\frac{1}{\sigma_g^2}X^T\Lambda\right)\hat{\Sigma}_{g21}\right) \\ & - \frac{1}{2}tr\left(\left(\frac{1}{\sigma_g^2}\Lambda^T\Lambda + I_q\right)\hat{\Sigma}_{g22}\right). \end{aligned} \tag{B.17}$$

## M-step

For each parameter, the ML estimate is obtained via conditional maximization of $Q$ with respect to the parameter with the other parameters and E-step estimates held fixed at their current values. With $\left[\hat{\beta}_g\right]_1$ standing for the first element

of the vector $\hat{\beta}_g$, and $\left[\hat{\Sigma}_{g11}\right]_{11}$ the $(1,1)$ element of the $2\times 2$ matrix $\hat{\Sigma}_{g11}$, and so on,

$$
\begin{aligned}
\hat{\mu} &= \frac{1}{\sum_g 1/\sigma_g^2}\sum_g \frac{1}{\sigma_g^2}\left(y_g - X\hat{\beta}_g - X_c\gamma_g - \Lambda\hat{F}_g\right)\\
\hat{\Lambda} &= \left[\sum_g \frac{1}{\sigma_g^2}\left(\left(y_g - \mu - X\hat{\beta}_g - X_c\gamma_g\right)\hat{F}_g^T - 2\times\frac{1}{2}X\hat{\Sigma}_{g21}^T\right)\right]\left[\sum_g \frac{1}{\sigma_g^2}\left(\hat{F}_g\hat{F}_g^T + \hat{\Sigma}_{g22}\right)\right]^{-1}\\
\hat{\gamma}_g &= \left(X_c^T X_c\right)^{-1} X_c^T\left(y_g - \mu - X\hat{\beta}_g - \Lambda\hat{F}_g\right)\\
\hat{\sigma}_0^2 &= \frac{1}{G}\sum_g\left(\left[\hat{\beta}_g\right]_1^2 + \left[\hat{\Sigma}_{g11}\right]_{11}\right)\\
\hat{\sigma}_1^2 &= \frac{1}{G}\sum_g\left(\left[\hat{\beta}_g\right]_2^2 + \left[\hat{\Sigma}_{g11}\right]_{22}\right)\\
\hat{\sigma}_g^2 &= \frac{1}{n}\left(\left\|y_g - \mu - X_c\gamma_g - X\hat{\beta}_g - \Lambda\hat{F}_g\right\|^2\right.\\
&\qquad \left. + tr\left(X^T X\hat{\Sigma}_{g11}\right) + 2\times tr\left(X^T\Lambda\hat{\Sigma}_{g21}\right) + tr\left(\Lambda^T\Lambda\hat{\Sigma}_{g22}\right)\right).
\end{aligned}
\tag{B.18}
$$

## Iterative Scheme

To summarize, the main steps of the algorithm are

i). Initialize $\Theta = \left\{\mu, \Lambda, \{\gamma_g\}, \sigma_0^2, \sigma_1^2, \{\sigma_g^2\}\right\}$.

ii). Update missing data in E-step (B.16) conditional on current values of $\Theta$.

iii). Update $\Theta$ in M-step (B.18) conditional on current value of missing data.

iv). Iterate cycles consisting of ii) and iii) until the increase in $Q = \sum_g Q_g$ (B.17) is negligible.

### Initial values

We use the same initial value for $\mu, \Lambda, \sigma_0^2$, and $\sigma_1^2$ as in RRmix. Also, we use initialize $\{\sigma_g^2\}$ to its initial shrinkage estimate in RRmix and $\{\gamma_g\}$ to a zero matrix.

**Calculation of p-values**

For each gene $g$, we construct the following posterior t-statistic

$$\frac{\left[\hat{\beta}_g\right]_2}{\sqrt{\left[\hat{\Sigma}_{g11}\right]_{22}}}$$

with each parameter at its estimated value upon convergence of the EM algorithm. Then p-values are computed under the standard normal distribution.

## B.5 RR model and EM algorithm

For any gene $g$, the model is the same as the RRmix model, except that there is no mixture structure

$$\begin{aligned}
y_g &= \mu + X_c\gamma_g + X\beta_g + \Lambda F_g + W_g \\
W_g &\sim N_n(0, \sigma_g^2 I_n) \\
F_g &\sim N_q(0, I_q) \\
\sigma_g^2 &\sim IG(A, B) \\
\beta_g &\sim N_2\left(0\,,\, \begin{bmatrix} \sigma_0^2 & 0 \\ 0 & \sigma_1^2 \end{bmatrix}\right).
\end{aligned}$$

Parameters to be estimated: $\Theta = \left\{\mu, \Lambda, \{\gamma_g\}, \sigma_0^2, \sigma_1^2\right\}$.

Shrinkage estimate to be calculated: $\{\sigma_g^2\}$.

Missing data: $\left\{\{\beta_g\}, \{F_g\}\right\}$.

## Similarity to the RF model

The estimation formulas in E- and M-steps and the $Q$ function are the same as in the RF model. However, $\{\sigma_g^2\}$ is now estimated via shrinkage estimation. Shrinkage estimation of error variance $\{\sigma_g^2\}$ follows the same procedure as described in the algorithm of the RRmix model.

## Iterative Scheme

To summarize, the main steps of the algorithm are

i). Initialize $\Theta = \left\{\mu, \Lambda, \{\gamma_g\}, \sigma_0^2, \sigma_1^2\right\}$.

ii). Initialize shrinkage estimate of $\{\sigma_g^2\}$ (B.11).

iii). Update missing data in E-step (B.16) conditional on current value of $\Theta$ and shrinkage estimate of $\{\sigma_g^2\}$.

iv). Update $\Theta$ in M-step (B.18) conditional on current value of missing data and shrinkage estimate of $\{\sigma_g^2\}$.

v). Iterate cycles consisting of ii), iii) and iv) until the increase in $Q = \sum_g Q_g$ (B.17) is negligible.

### Initial values

We use the same initial value for $\mu, \Lambda, \sigma_0^2$, and $\sigma_1^2$ as in RRmix. Also, we use initialize $\{\sigma_g^2\}$ to its initial shrinkage estimate in RRmix and $\{\gamma_g\}$ to a zero matrix.

**Calculation of p-values**

p-values are calculated in the same way as in the RF model, under the standard normal distribution. Similar to how FF and FR models differ, the difference in p-values and in turn in the outcome of the multiple hypothesis tests between the RF and RR models lies in that RF models involves ML estimate of the error variance $\{\sigma_g^2\}$ whereas RR model involves shrinkage estimate of the error variance.

# APPENDIX C

## SUPPLEMENTARY INFORMATION OF CHAPTER 5

Table C.1: List of top 100 non-duplicated eQTL associations identified by FR, with a Bonferroni corrected significance level $0.05/(7,575 \times 191,959)$. The columns from left to right are ranking, Ensemble code for the genes, SNP chromosome, SNP position, p-value of gene being non-null with respect to SNP, and TRUE/FALSE (T/F) for whether the gene-SNP association is cis. There are 64 cis associations in the list.

| Ranking | GeneID | Chr (SNP) | Position | SNP | P-value | IsCis |
|---|---|---|---|---|---|---|
| 1 | 6840 | 0 | 0 | rs3858231 | 0.000E+00 | F |
| 2 | 23421 | 1 | 63759453 | rs17391823 | 0.000E+00 | T |
| 3 | 5906 | 1 | 112038561 | rs1886498 | 0.000E+00 | T |
| 4 | 54879 | 1 | 112933916 | rs6666579 | 0.000E+00 | T |
| 5 | 100506015 | 2 | 161907239 | rs10197817 | 0.000E+00 | T |
| 6 | 164781 | 2 | 228485182 | rs3748863 | 0.000E+00 | T |
| 7 | 5947 | 3 | 140736561 | rs12485273 | 0.000E+00 | F |
| 8 | 25854 | 4 | 187345974 | rs4586997 | 0.000E+00 | F |
| 9 | 1915 | 6 | 74287580 | rs3822960 | 0.000E+00 | T |
| 10 | 26503 | 6 | 74395913 | rs9446955 | 0.000E+00 | T |
| 11 | 80177 | 6 | 152990802 | rs7765586 | 0.000E+00 | T |
| 12 | 5340 | 6 | 161057248 | rs4252106 | 0.000E+00 | T |
| 13 | 90637 | 7 | 1170367 | rs2949192 | 0.000E+00 | T |
| 14 | 155368 | 7 | 72863628 | rs4355658 | 0.000E+00 | T |
| 15 | 7976 | 8 | 28429267 | rs2874941 | 0.000E+00 | T |
| 16 | 8000 | 8 | 143761003 | rs2976396 | 0.000E+00 | F |
| 17 | 84545 | 10 | 102724768 | rs4919510 | 0.000E+00 | F |
| 18 | 80150 | 11 | 61840106 | rs1406384 | 0.000E+00 | T |
| 19 | 374491 | 13 | 24078151 | rs943049 | 0.000E+00 | T |
| 20 | 91612 | 14 | 64477410 | rs2412065 | 0.000E+00 | T |
| 21 | 4649 | 15 | 69903832 | rs2742323 | 0.000E+00 | F |
| 22 | 146562 | 16 | 4734874 | rs2075469 | 0.000E+00 | T |
| 23 | 63928 | 16 | 23697839 | rs194788 | 0.000E+00 | T |
| 24 | 5889 | 17 | 54043788 | rs1864754 | 0.000E+00 | F |
| 25 | 388407 | 17 | 56847496 | rs758598 | 0.000E+00 | F |
| 26 | 55034 | 18 | 31980483 | rs3737468 | 0.000E+00 | F |
| 27 | 5268 | 18 | 59295033 | rs3744941 | 0.000E+00 | F |
| 28 | 89778 | 18 | 59523027 | rs9960039 | 0.000E+00 | F |
| 29 | 26090 | 20 | 25166618 | rs6132819 | 0.000E+00 | T |
| 30 | 150142 | 21 | 42317483 | rs220219 | 0.000E+00 | T |
| 31 | 538 | 23 | 77217818 | rs2643591 | 0.000E+00 | T |
| 32 | 140686 | 20 | 43836276 | rs2664529 | 1.267E-55 | F |
| 33 | 114757 | 17 | 72058534 | rs752049 | 1.205E-38 | F |
| 34 | 388335 | 17 | 10556076 | rs397278 | 6.201E-33 | T |
| 35 | 340542 | 23 | 101168380 | rs2858353 | 1.240E-30 | T |
| 36 | 116285 | 16 | 20587707 | rs433598 | 3.143E-29 | T |
| 37 | 51144 | 11 | 43796511 | rs10768983 | 2.247E-27 | T |
| 38 | 26751 | 2 | 217201 | rs17713396 | 6.917E-27 | T |
| 39 | 403314 | 1 | 181889002 | rs6699011 | 7.785E-27 | F |
| 40 | 318 | 9 | 34271390 | rs7045680 | 7.987E-27 | T |

Table C.1 – Continued from previous page

| Ranking | GeneID | Chr (SNP) | Position | SNP | Post.Prob. | IsCis |
|---|---|---|---|---|---|---|
| 41 | 158158 | 9 | 84802863 | rs6559694 | 1.462E-25 | F |
| 42 | 84930 | 10 | 27494117 | rs7068375 | 4.310E-23 | T |
| 43 | 25961 | 10 | 74540513 | rs2280369 | 6.884E-23 | F |
| 44 | 286464 | 23 | 36068724 | rs16987374 | 2.355E-22 | T |
| 45 | 55278 | 6 | 107221054 | rs1026619 | 2.767E-22 | T |
| 46 | 22948 | 5 | 10318076 | rs699113 | 3.058E-22 | T |
| 47 | 10230 | 17 | 38773860 | rs4793229 | 4.893E-22 | F |
| 48 | 121506 | 12 | 14961902 | rs2193356 | 8.929E-21 | T |
| 49 | 1965 | 14 | 66916971 | rs8008724 | 1.261E-20 | T |
| 50 | 100507580 | 22 | 24215353 | rs6004673 | 2.048E-19 | F |
| 51 | 10781 | 19 | 9394805 | rs6512121 | 4.950E-18 | T |
| 52 | 10558 | 9 | 93804054 | rs7045602 | 7.063E-18 | F |
| 53 | 100131564 | 1 | 93561736 | rs7555292 | 7.508E-18 | T |
| 54 | 54960 | 23 | 13929093 | rs7055913 | 2.751E-17 | T |
| 55 | 26999 | 5 | 156687851 | rs13155266 | 7.289E-17 | T |
| 56 | 8624 | 21 | 39555501 | rs2836965 | 7.321E-17 | T |
| 57 | 6263 | 15 | 31881493 | rs2115747 | 9.108E-17 | F |
| 58 | 2882 | 1 | 52836600 | rs835341 | 1.098E-16 | T |
| 59 | 100507540 | 9 | 74103814 | rs7874628 | 2.220E-16 | F |
| 60 | 55020 | 22 | 45066872 | rs6008552 | 4.441E-16 | F |
| 61 | 27030 | 14 | 74615438 | rs175490 | 5.639E-16 | T |
| 62 | 100507316 | 8 | 144410365 | rs7824894 | 7.337E-16 | F |
| 63 | 51531 | 9 | 99670770 | rs7357707 | 7.429E-16 | F |
| 64 | 51816 | 22 | 16062294 | rs1076106 | 1.123E-15 | F |
| 65 | 54847 | 3 | 114737833 | rs4580515 | 1.462E-15 | F |
| 66 | 55728 | 4 | 39796815 | rs17619330 | 1.637E-15 | T |
| 67 | 55253 | 7 | 66219599 | rs17144722 | 1.776E-15 | T |
| 68 | 151525 | 2 | 159886945 | rs174227 | 1.998E-15 | T |
| 69 | 79772 | 5 | 93992505 | rs10052066 | 2.369E-15 | T |
| 70 | 145957 | 15 | 73938020 | rs2593280 | 2.593E-15 | F |
| 71 | 201651 | 3 | 152973681 | rs4679934 | 3.109E-15 | F |
| 72 | 100506707 | 2 | 113705738 | rs2863243 | 4.931E-15 | T |
| 73 | 654433 | 2 | 113700504 | rs3748916 | 6.439E-15 | T |
| 74 | 10783 | 9 | 126069557 | rs12379417 | 9.190E-15 | F |
| 75 | 51016 | 14 | 23681818 | rs3742500 | 1.161E-14 | T |
| 76 | 64105 | 5 | 64951379 | rs2161278 | 1.188E-14 | T |
| 77 | 51703 | 10 | 114138679 | rs12255316 | 1.430E-14 | F |
| 78 | 79618 | 8 | 28881393 | rs4732896 | 1.613E-14 | T |
| 79 | 143241 | 10 | 82062290 | rs10788562 | 2.271E-14 | F |
| 80 | 622 | 3 | 198745276 | rs13077136 | 2.598E-14 | F |
| 81 | 133383 | 5 | 56214969 | rs832584 | 5.309E-14 | T |
| 82 | 643529 | 10 | 91582178 | rs1125326 | 7.483E-14 | F |
| 83 | 57545 | 4 | 15091907 | rs6810461 | 8.297E-14 | T |
| 84 | 55198 | 12 | 104142992 | rs2440695 | 1.134E-13 | F |
| 85 | 55733 | 1 | 208584705 | rs6696657 | 1.432E-13 | F |
| 86 | 84221 | 21 | 46524284 | rs2839195 | 1.684E-13 | T |
| 87 | 669 | 7 | 133960652 | rs1160049 | 1.767E-13 | T |
| 88 | 55234 | 9 | 33035161 | rs10758181 | 1.872E-13 | T |
| 89 | 7317 | 4 | 48077300 | rs3943122 | 2.274E-13 | T |
| 90 | 7174 | 13 | 102123999 | rs660236 | 2.303E-13 | T |
| 91 | 6006 | 1 | 25641524 | rs10903129 | 2.585E-13 | T |
| 92 | 3631 | 2 | 98450324 | rs12617721 | 2.611E-13 | T |
| 93 | 11005 | 5 | 147449164 | rs4529181 | 2.971E-13 | T |
| 94 | 401491 | 9 | 2527815 | rs588933 | 3.279E-13 | T |

Table C.1 – Continued from previous page

| Ranking | GeneID | Chr (SNP) | Position | SNP | Post.Prob. | IsCis |
|---|---|---|---|---|---|---|
| 95 | 80868 | 6 | 30024484 | rs417162 | 3.559E-13 | T |
| 96 | 55125 | 18 | 12967206 | rs8088313 | 4.206E-13 | T |
| 97 | 10521 | 22 | 37237795 | rs138465 | 4.372E-13 | F |
| 98 | 286042 | 8 | 8185680 | rs2945886 | 4.841E-13 | T |
| 99 | 6716 | 2 | 31620635 | rs4952197 | 4.863E-13 | T |
| 100 | 7180 | 6 | 49811816 | rs597544 | 5.462E-13 | T |

Table C.2: List of top 100 non-duplicated eQTL associations identified by RRmix, with a threshold of 0.9999 on the posterior non-null probability. The columns from left to right are ranking, Ensemble code for the genes, SNP chromosome, SNP position, posterior probability of gene being non-null with respect to SNP, and TRUE/FALSE (T/F) for whether the gene-SNP association is cis. There are 54 cis associations in the list.

| Ranking | GeneID | Chr (SNP) | Position | SNP | Post.Prob. | IsCis |
|---|---|---|---|---|---|---|
| 1 | 6840 | 0 | 0 | rs3858231 | 1.0000000000 | F |
| 2 | 23421 | 1 | 63759453 | rs17391823 | 1.0000000000 | T |
| 3 | 5906 | 1 | 112038561 | rs1886498 | 1.0000000000 | T |
| 4 | 403314 | 1 | 181889002 | rs6699011 | 1.0000000000 | F |
| 5 | 26751 | 2 | 217201 | rs17713396 | 1.0000000000 | T |
| 6 | 6192 | 2 | 156905829 | rs707130 | 1.0000000000 | F |
| 7 | 164781 | 2 | 228485182 | rs3748863 | 1.0000000000 | T |
| 8 | 5947 | 3 | 140736561 | rs12485273 | 1.0000000000 | F |
| 9 | 22948 | 5 | 10300965 | rs2548553 | 1.0000000000 | T |
| 10 | 1915 | 6 | 74287580 | rs3822960 | 1.0000000000 | T |
| 11 | 26503 | 6 | 74395913 | rs9446955 | 1.0000000000 | T |
| 12 | 80177 | 6 | 152990802 | rs7765586 | 1.0000000000 | T |
| 13 | 5340 | 6 | 161057248 | rs4252106 | 1.0000000000 | T |
| 14 | 90637 | 7 | 1170367 | rs2949192 | 1.0000000000 | T |
| 15 | 155368 | 7 | 72863628 | rs4355658 | 1.0000000000 | T |
| 16 | 7976 | 8 | 28429267 | rs2874941 | 1.0000000000 | T |
| 17 | 8000 | 8 | 143761003 | rs2976396 | 1.0000000000 | F |
| 18 | 318 | 9 | 34271390 | rs7045680 | 1.0000000000 | T |
| 19 | 158158 | 9 | 84791616 | rs4385545 | 1.0000000000 | F |
| 20 | 84930 | 10 | 27494117 | rs7068375 | 1.0000000000 | T |
| 21 | 25961 | 10 | 74540513 | rs2280369 | 1.0000000000 | F |
| 22 | 51144 | 11 | 43710301 | rs6485455 | 1.0000000000 | T |
| 23 | 374491 | 13 | 24078151 | rs943049 | 1.0000000000 | T |
| 24 | 91612 | 14 | 64477410 | rs2412065 | 1.0000000000 | T |
| 25 | 1965 | 14 | 66916971 | rs8008724 | 1.0000000000 | T |
| 26 | 116285 | 16 | 20587707 | rs433598 | 1.0000000000 | T |
| 27 | 63928 | 16 | 23697839 | rs194788 | 1.0000000000 | T |
| 28 | 388335 | 17 | 10552154 | rs483734 | 1.0000000000 | T |
| 29 | 114757 | 17 | 72053478 | rs895156 | 1.0000000000 | F |

Table C.2 – Continued from previous page

| Ranking | GeneID | Chr (SNP) | Position | SNP | Post.Prob. | IsCis |
|---|---|---|---|---|---|---|
| 30 | 5268 | 18 | 59295033 | rs3744941 | 1.0000000000 | F |
| 31 | 89778 | 18 | 59523027 | rs9960039 | 1.0000000000 | F |
| 32 | 140686 | 20 | 43836276 | rs2664529 | 1.0000000000 | F |
| 33 | 150142 | 21 | 42317483 | rs220219 | 1.0000000000 | T |
| 34 | 100507580 | 22 | 24204464 | rs6004667 | 1.0000000000 | F |
| 35 | 286464 | 23 | 36060001 | rs5973583 | 1.0000000000 | T |
| 36 | 340542 | 23 | 101168380 | rs2858353 | 1.0000000000 | T |
| 37 | 55278 | 6 | 107221054 | rs1026619 | 1.0000000000 | T |
| 38 | 80150 | 11 | 61840106 | rs1406384 | 1.0000000000 | T |
| 39 | 100506015 | 2 | 161907239 | rs10197817 | 1.0000000000 | T |
| 40 | 84545 | 10 | 102724768 | rs4919510 | 1.0000000000 | F |
| 41 | 284323 | 19 | 45181310 | rs8105066 | 1.0000000000 | F |
| 42 | 26090 | 20 | 25223843 | rs2258719 | 1.0000000000 | T |
| 43 | 8653 | 12 | 64263570 | rs7308337 | 1.0000000000 | F |
| 44 | 121506 | 12 | 14961902 | rs2193356 | 1.0000000000 | T |
| 45 | 10781 | 19 | 9376200 | rs4804436 | 1.0000000000 | T |
| 46 | 7503 | 15 | 77022501 | rs3825932 | 1.0000000000 | F |
| 47 | 8287 | 12 | 64263570 | rs7308337 | 1.0000000000 | F |
| 48 | 100131564 | 1 | 93561736 | rs7555292 | 1.0000000000 | T |
| 49 | 10783 | 9 | 126069557 | rs12379417 | 1.0000000000 | F |
| 50 | 54879 | 1 | 112933916 | rs6666579 | 1.0000000000 | T |
| 51 | 84221 | 21 | 46524284 | rs2839195 | 1.0000000000 | T |
| 52 | 2882 | 1 | 52836600 | rs835341 | 1.0000000000 | T |
| 53 | 55020 | 22 | 45066872 | rs6008552 | 1.0000000000 | F |
| 54 | 25854 | 4 | 187345974 | rs4586997 | 1.0000000000 | F |
| 55 | 55034 | 18 | 31980483 | rs3737468 | 1.0000000000 | F |
| 56 | 143241 | 10 | 82062290 | rs10788562 | 1.0000000000 | F |
| 57 | 538 | 23 | 77217818 | rs2643591 | 1.0000000000 | T |
| 58 | 26999 | 5 | 156687851 | rs13155266 | 1.0000000000 | T |
| 59 | 388407 | 17 | 56851431 | rs2079795 | 1.0000000000 | F |
| 60 | 654433 | 2 | 113700504 | rs3748916 | 1.0000000000 | T |
| 61 | 51816 | 22 | 16062294 | rs1076106 | 1.0000000000 | F |
| 62 | 643529 | 10 | 91576021 | rs7083051 | 1.0000000000 | F |
| 63 | 4649 | 15 | 69903832 | rs2742323 | 1.0000000000 | F |
| 64 | 64105 | 5 | 64951379 | rs2161278 | 1.0000000000 | T |
| 65 | 10230 | 17 | 38773860 | rs4793229 | 1.0000000000 | F |
| 66 | 8284 | 1 | 57402866 | rs12059674 | 1.0000000000 | F |
| 67 | 8624 | 21 | 39555501 | rs2836965 | 1.0000000000 | T |
| 68 | 27030 | 14 | 74615438 | rs175490 | 1.0000000000 | T |
| 69 | 283078 | 10 | 28057578 | rs7358151 | 1.0000000000 | T |
| 70 | 6263 | 15 | 31881493 | rs2115747 | 1.0000000000 | F |
| 71 | 201651 | 3 | 152973681 | rs4679934 | 1.0000000000 | F |
| 72 | 9086 | 6 | 151114691 | rs7774822 | 1.0000000000 | F |
| 73 | 55125 | 18 | 12967206 | rs8088313 | 1.0000000000 | T |
| 74 | 286042 | 8 | 8185680 | rs2945886 | 0.9999999999 | T |
| 75 | 145957 | 15 | 73938020 | rs2593280 | 0.9999999999 | F |
| 76 | 10558 | 9 | 93804054 | rs7045602 | 0.9999999999 | F |
| 77 | 100507540 | 9 | 74103814 | rs7874628 | 0.9999999999 | F |
| 78 | 83606 | 12 | 73771142 | rs10879885 | 0.9999999999 | F |
| 79 | 51531 | 9 | 99670770 | rs7357707 | 0.9999999999 | F |
| 80 | 151525 | 2 | 159886945 | rs174227 | 0.9999999999 | T |
| 81 | 8458 | 1 | 117401993 | rs12029623 | 0.9999999999 | T |
| 82 | 51703 | 10 | 114138679 | rs12255316 | 0.9999999998 | F |
| 83 | 55253 | 7 | 66219599 | rs17144722 | 0.9999999998 | T |

Table C.2 – Continued from previous page

| Ranking | GeneID | Chr (SNP) | Position | SNP | Post.Prob. | IsCis |
|---|---|---|---|---|---|---|
| 84 | 146562 | 16 | 4734874 | rs2075469 | 0.9999999997 | T |
| 85 | 8987 | 4 | 77415524 | rs1441914 | 0.9999999997 | F |
| 86 | 55728 | 4 | 39796815 | rs17619330 | 0.9999999996 | T |
| 87 | 83439 | 2 | 85355930 | rs12714138 | 0.9999999996 | T |
| 88 | 5889 | 17 | 54195586 | rs8074016 | 0.9999999996 | F |
| 89 | 622 | 3 | 198745276 | rs13077136 | 0.9999999995 | F |
| 90 | 79772 | 5 | 93992505 | rs10052066 | 0.9999999995 | T |
| 91 | 100506707 | 2 | 113705738 | rs2863243 | 0.9999999994 | T |
| 92 | 10521 | 22 | 37237795 | rs138465 | 0.9999999993 | F |
| 93 | 144402 | 12 | 37325525 | rs2653762 | 0.9999999991 | F |
| 94 | 285533 | 8 | 116142633 | rs6469567 | 0.9999999989 | F |
| 95 | 123 | 14 | 26210860 | rs1950420 | 0.9999999985 | F |
| 96 | 2628 | 15 | 43471087 | rs1145077 | 0.9999999982 | F |
| 97 | 54960 | 23 | 13929093 | rs7055913 | 0.9999999982 | T |
| 98 | 51016 | 14 | 23681818 | rs3742500 | 0.9999999981 | T |
| 99 | 84663 | 12 | 64263570 | rs7308337 | 0.9999999977 | F |
| 100 | 57545 | 4 | 15091907 | rs6810461 | 0.9999999975 | T |

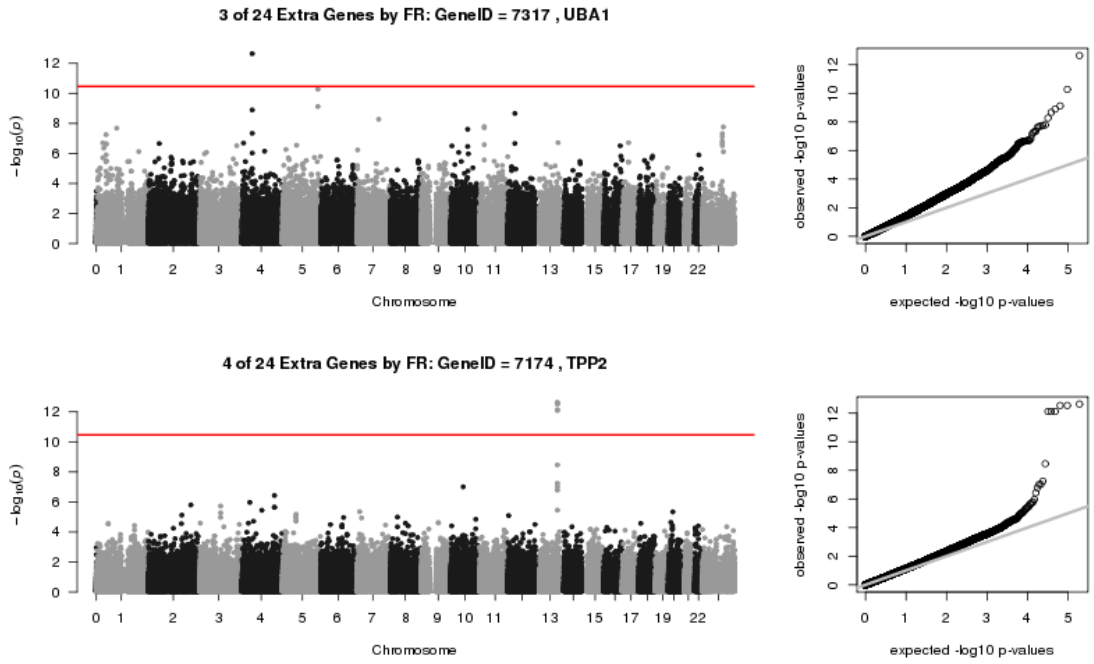


Figure C.1: Manhattan and QQ plots for the third and fourth (TPP2) genes for which a significant cis association is identified by FR but not by HEFT. Gene ranking is based on ascending p-value among the 24 extra cis associations. A reference line at $-\log_{10}(0.05/(7,575 \times 191,959)) = 10.46$ for the Bonforroni threshold is indicated in the Manhattan plots. Deviation from the reference line in the QQ plot and lack of unique peak in the Manhattan plots suggest that the cis association detected by FR for gene UBA1 (top) with gene ID 7317 is likely a false positive.

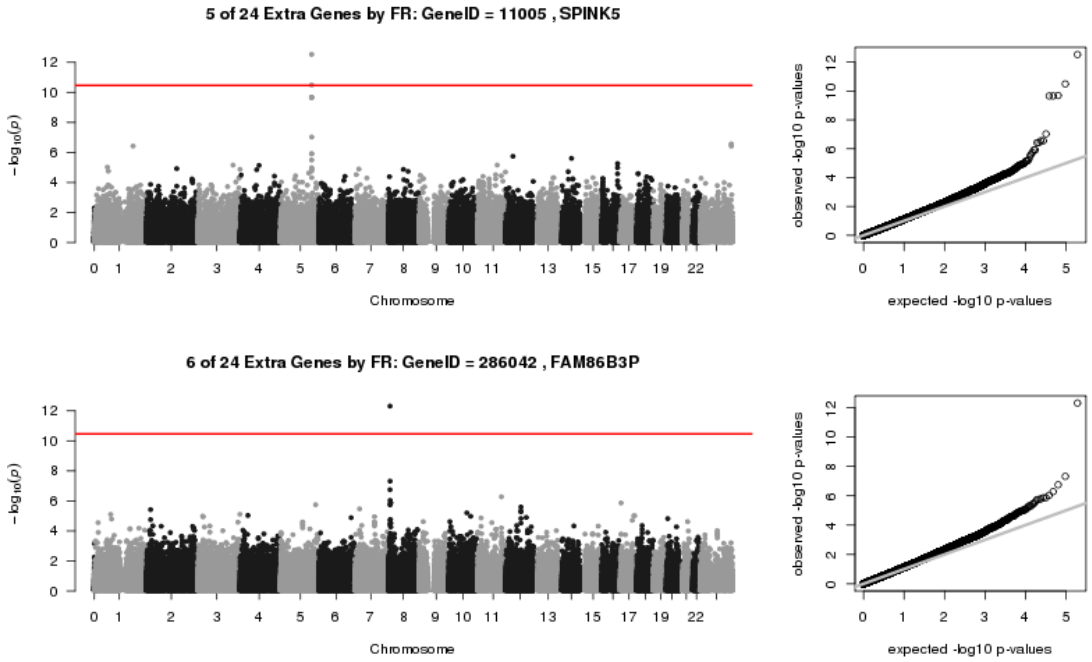


Figure C.2: Manhattan and QQ plots for the fifth (SPINK5) and sixth genes for which a significant cis association is identified by FR but not by HEFT. Gene ranking is based on ascending p-value among the 24 extra cis associations. A reference line at $-\log_{10}$ of the Bonforroni threshold is indicated in the Manhattan plots.

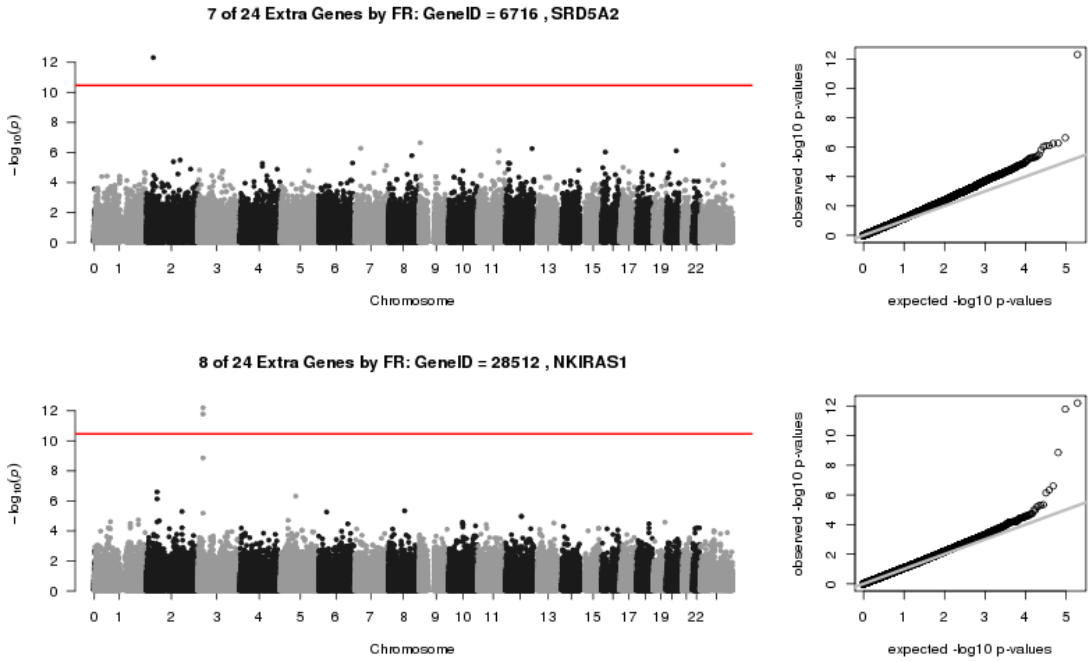


Figure C.3: Manhattan and QQ plots for the seventh and eighth (NKIRAS1) genes for which a significant cis association is identified by FR but not by HEFT. Gene ranking is based on ascending p-value among the 24 extra cis associations. A reference line at $-\log_{10}$ of the Bonforroni threshold is indicated in the Manhattan plots.

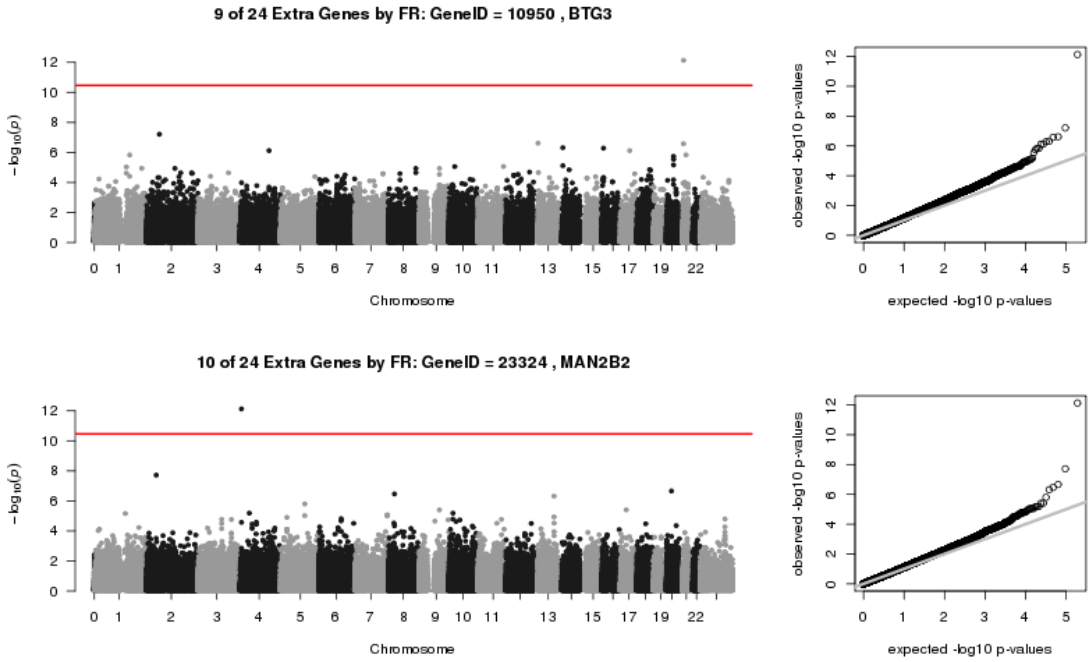


Figure C.4: Manhattan and QQ plots for the ninth (BTG3) and tenth genes for which a significant cis association is identified by FR but not by HEFT. Gene ranking is based on ascending p-value among the 24 extra cis associations. A reference line at $-\log_{10}$ of the Bonforroni threshold is indicated in the Manhattan plots.

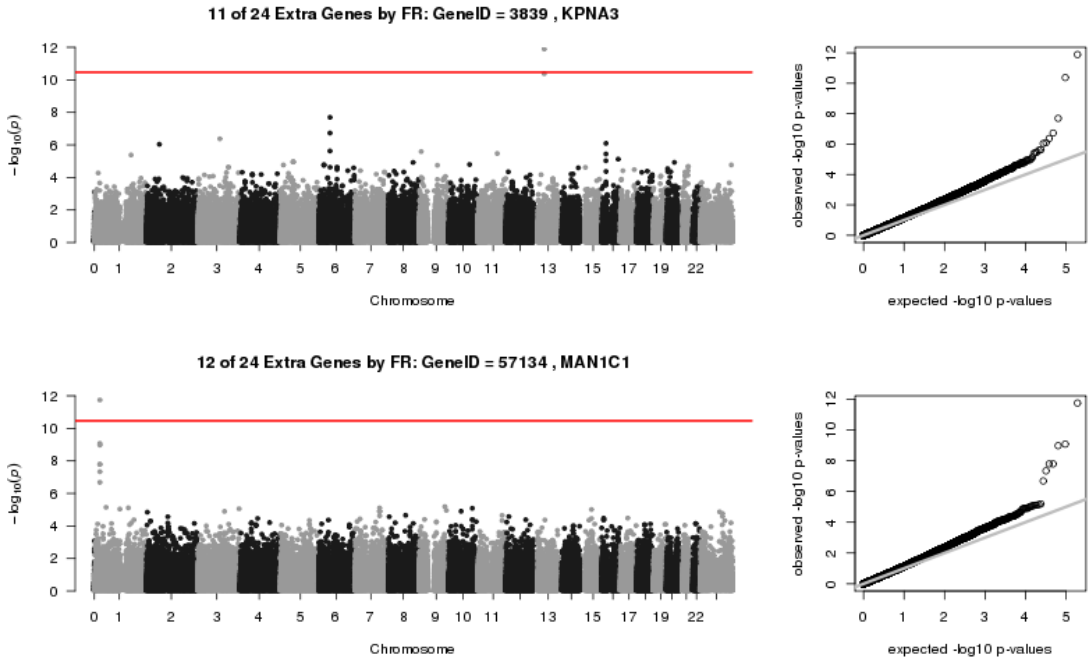


Figure C.5: Manhattan and QQ plots for the eleventh and twelfth (MAN1C1) genes for which a significant cis association is identified by FR but not by HEFT. Gene ranking is based on ascending p-value among the 24 extra cis associations. A reference line at $-\log_{10}$ of the Bonforroni threshold is indicated in the Manhattan plots.

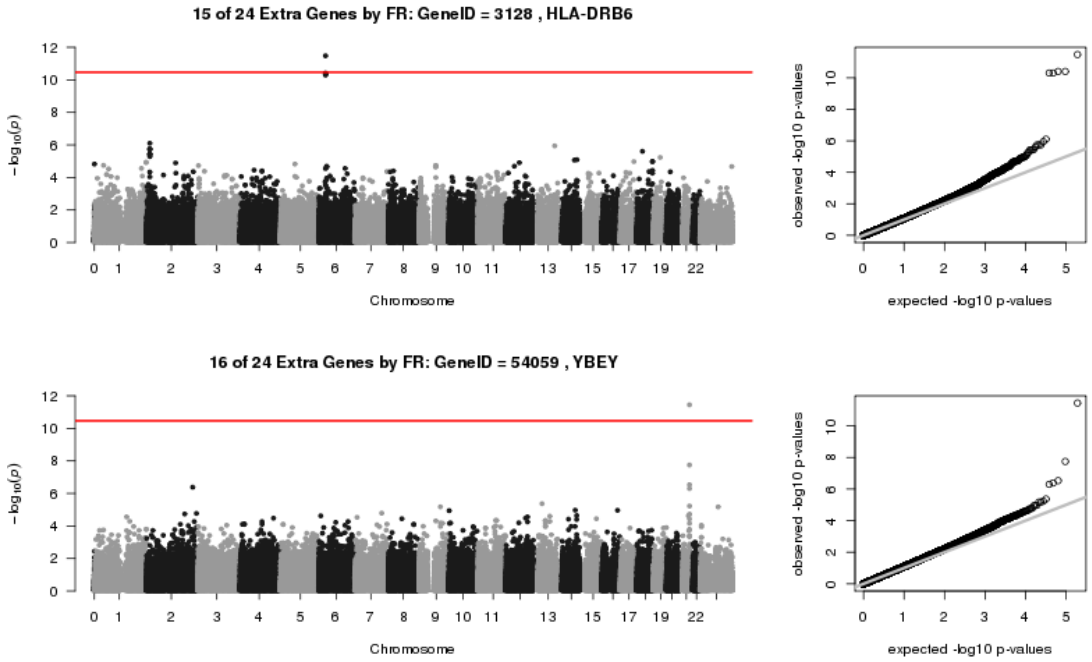


Figure C.6: Manhattan and QQ plots for the fifteenth and sixteenth (YBEY) genes for which a significant cis association is identified by FR but not by HEFT. Gene ranking is based on ascending p-value among the 24 extra cis associations. A reference line at $-\log_{10}$ of the Bonforroni threshold is indicated in the Manhattan plots.

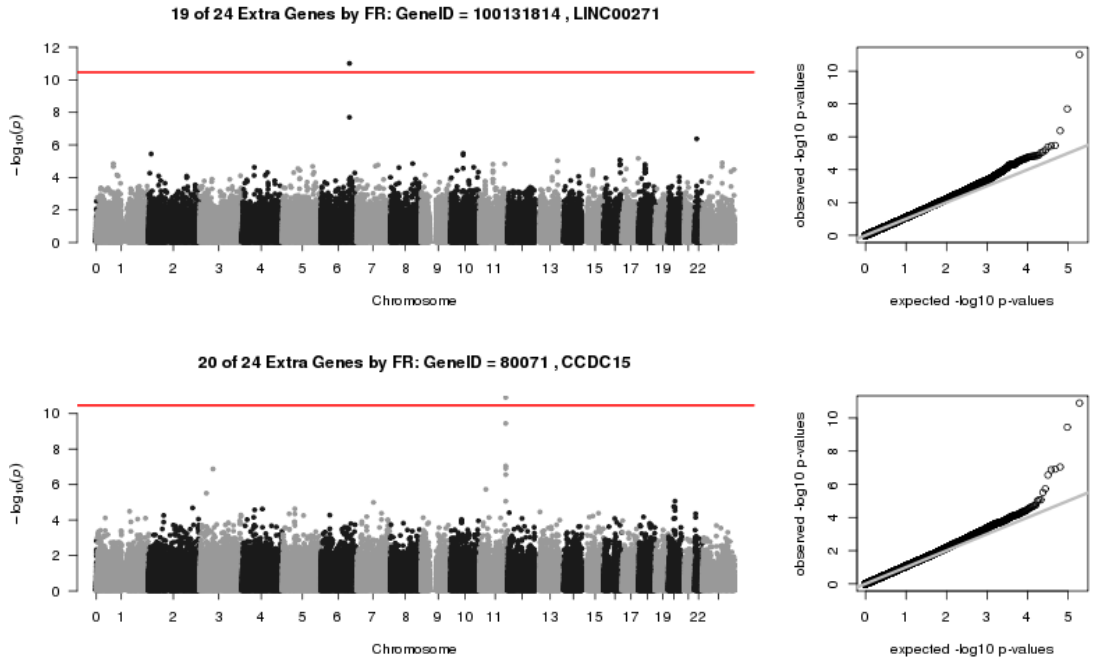


Figure C.7: Manhattan and QQ plots for the nineteenth and twentieth (CCDC15) genes for which a significant cis association is identified by FR but not by HEFT. Gene ranking is based on ascending p-value among the 24 extra cis associations. A reference line at $-\log_{10}$ of the Bonforroni threshold is indicated in the Manhattan plots.

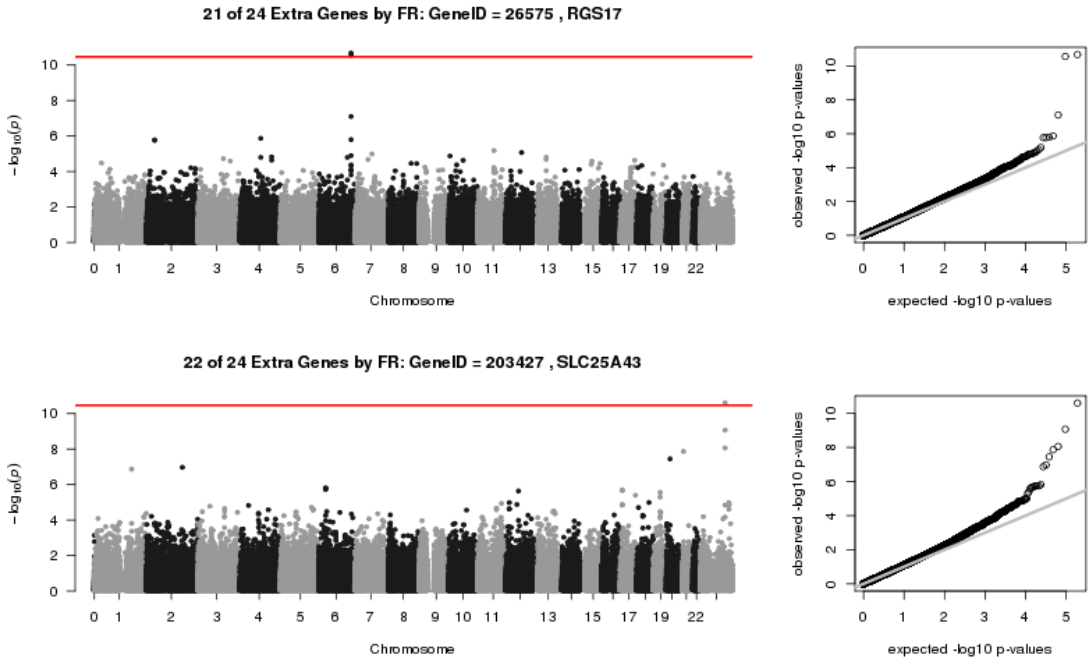


Figure C.8: Manhattan and QQ plots for the twenty-first (RGS17) and twenty-second genes for which a significant cis association is identified by FR but not by HEFT. Gene ranking is based on ascending p-value among the 24 extra cis associations. A reference line at $-\log_{10}$ of the Bonforroni threshold is indicated in the Manhattan plots.

## BIBLIOGRAPHY

Accordini, S., Bombieri, C., Malerba, G., Belpinati, F., Calciano, L., Ferrari, M., Presti, A. R. L., Perbellini, L., Pesce, G., Pignatti, P. F., et al. (2013). SNPs in SPINK5 gene region are associated with asthma severity: Preliminary results from the gene environment interactions in respiratory diseases (GEIRD) study. *European Respiratory Journal*, 42(Suppl 57):1524.

Alon, U., Barkai, N., Notterman, D., Gish, K., Ybarra, S., Mack, D., and Levine, A. (1999). Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays. *Proceedings of the National Academy of Sciences*, 96(12):6745–6750.

Attias, H. (2000). A variational Bayesian framework for graphical models. *Advances in Neural Information Processing Systems*, 12(1-2):209–215.

Bar, H., Booth, J., Schifano, E., and Wells, M. (2010). Laplace approximated EM microarray analysis: an empirical Bayes approach for comparative microarray experiments. *Statistical Science*, 25(3):388–407.

Bar, H. and Schifano, E. (2010). Lemma: Laplace approximated EM microarray analysis. R package version 1.3-1. `http://CRAN.R-project.org/package=lemma`.

Beal, M. (2003). *Variational Algorithms for Approximate Bayesian Inference*. PhD thesis, University of London.

Bishop, C. (1999). Variational principal components. In *Proceedings of Ninth International Conference on Artificial Neural Networks, ICANN'99*, volume 1, pages 509–514. IET.

Bishop, C. (2006). *Pattern Recognition and Machine Learning*. Springer Science+ Business Media, New York.

Bishop, C., Spiegelhalter, D., and Winn, J. (2002). VIBES: A variational inference engine for Bayesian networks. *Advances in Neural Information Processing Systems*, 15:777–784.

Blalock, E. M., Geddes, J. W., Chen, K. C., Porter, N. M., Markesbery, W. R., and Landfield, P. W. (2004). Incipient Alzheimer's disease: microarray correlation analyses reveal major transcriptional and tumor suppressor responses. *Proceedings of the National Academy of Sciences of the United States of America*, 101(7):2173–2178.

Blei, D. and Jordan, M. (2006). Variational inference for Dirichlet process mixtures. *Bayesian Analysis*, 1(1):121–143.

Booth, J., Eilertson, K., Olinares, P., and Yu, H. (2011). A Bayesian mixture model for comparative spectral count data in shotgun proteomics. *Molecular & Cellular Proteomics*, 10(8).

Boyd, S. and Vandenberghe, L. (2004). *Convex Optimization*. Cambridge University Press.

Callow, M., Dudoit, S., Gong, E., Speed, T., and Rubin, E. (2000). Microarray expression profiling identifies genes with altered expression in HDL-deficient mice. *Genome Research*, 10(12):2022–2029.

Chen, X., Chen, G., Cao, X., Zhou, Y., Yang, T., and Wei, S. (2013). Downregulation of BTG3 in non-small cell lung cancer. *Biochemical and biophysical research communications*, 437(1):173–178.

Christensen, R., Johnson, W. O., Branscum, A. J., and Hanson, T. E. (2011). *Bayesian Ideas and Data Analysis: An Introduction for Scientists and Statisticians*. CRC Press, Boca Raton.

Consonni, G. and Marin, J. (2007). Mean-field variational approximate Bayesian inference for latent variable models. *Computational Statistics & Data Analysis*, 52(2):790–798.

Corduneanu, A. and Bishop, C. (2001). Variational Bayesian model selection for mixture distributions. In *Artificial Intelligence and Statistics 2001*, Tommi S. Jaakkola and Thomas S. Richardson (Eds.), pages 27–34. Morgan Kaufmann, Waltham.

Cowles, M. K. and Carlin, B. P. (1996). Markov chain Monte Carlo convergence diagnostics: a comparative review. *Journal of the American Statistical Association*, 91(434):883–904.

De Freitas, N., Højen-Sørensen, P., Jordan, M., and Russell, S. (2001). Variational MCMC. In *Proceedings of the Seventeenth Conference on Uncertainty in Artificial Intelligence*, John Breese and Daphne Koller (Eds.), pages 120–127. Morgan Kaufmann Publishers Inc., San Francisco.

Efron, B. (2008). Microarrays, empirical Bayes and the two-groups model. *Statistical Science*, 23(1):1–22.

Eisenberg, E. and Levanon, E. Y. (2003). Human housekeeping genes are compact. *TRENDS in Genetics*, 19(7):362–365.

Faes, C., Ormerod, J., and Wand, M. (2011). Variational Bayesian inference for parametric and nonparametric regression with missing data. *Journal of the American Statistical Association*, 106(495):959–971.

Friguet, C., Kloareg, M., and Causeur, D. (2009). A factor model approach to multiple testing under dependence. *Journal of the American Statistical Association*, 104(488):1406–1415.

Friston, K., Ashburner, J., Kiebel, S., Nichols, T., and Penny, W. (2011). *Statistical Parametric Mapping: The Analysis of Functional Brain Images*. Academic Press, London.

Fusi, N., Stegle, O., and Lawrence, N. D. (2012). Joint modelling of confounding factors and prominent genetic regulators provides increased accuracy in genetical genomics studies. *PLoS computational biology*, 8(1):e1002330.

Gagnon-Bartsch, J., Jacob, L., and Speed, T. (2013). Removing unwanted variation from high dimensional data with negative controls. Technical report, Technical Report 820, Department of Statistics, University of California, Berkeley.

Gagnon-Bartsch, J. A. and Speed, T. P. (2012). Using control genes to correct for unwanted variation in microarray data. *Biostatistics*, 13(3):539–552.

Gao, C., Tignor, N. L., Salit, J., Strulovici-Barel, Y., Hackett, N. R., Crystal, R. G., and Mezey, J. G. (2014). HEFT: eQTL analysis of many thousands of expressed genes while simultaneously controlling for hidden factors. *Bioinformatics*, 30(3):369–376.

Gelman, A., Carlin, J. B., Stern, H. S., and Rubin, D. B. (2003). *Bayesian Data Analysis*. Chapman & Hall/CRC.

Ghahramani, Z. and Beal, M. (2000). Variational inference for Bayesian mixtures of factor analysers. *Advances in Neural Information Processing Systems*, 12:449–455.

Goldsmith, J., Wand, M., and Crainiceanu, C. (2011). Functional regression via variational Bayes. *Electronic Journal of Statistics*, 5:572.

Golz, S., Bruggemeier, U., Geerts, A., and Summer, H. (2007). Diagnostics and therapeutics for diseases associated with Tripeptidyl-Peptidase 2(TPP2). US Patent App. 10/586,492.

Grimmer, J. (2011). An introduction to Bayesian inference via variational approximations. *Political Analysis*, 19(1):32–47.

Honkela, A. and Valpola, H. (2005). Unsupervised variational Bayesian learning of nonlinear models. In *Advances in Neural Information Processing Systems 17*, Lawrence K. Saul, Yair Weiss and Léon Bottou (Eds.), pages 593–600. MIT Press.

Jaakkola, T. S. (2000). Tutorial on variational approximation methods. In *Advanced Mean Field Methods: Theory and Practice*, Manfred Opper and David Saad (Eds.), pages 129–159. MIT Press.

Jacob, L., Gagnon-Bartsch, J., and Speed, T. P. (2012). Correcting gene expression data when neither the unwanted variation nor the factor of interest are observed. *arXiv preprint arXiv:1211.4259*.

Kendziorski, C., Chen, M., Yuan, M., Lan, H., and Attie, A. (2006). Statistical methods for expression quantitative trait loci (eQTL) mapping. *Biometrics*, 62(1):19–27.

Khodyrev, D., Pronina, I., Rykov, S., Beresneva, E., Freedman, M., Kazubskaya, T., Loginov, V., and Braga, E. (2012). Involvement of methylation of group of miRNA genes in regulation of expression of RAR-beta2 and NKIRAS1 target genes in lung cancer. *Molecular Biology*, 46(5):693–704.

Leek, J. T. and Storey, J. D. (2007). Capturing heterogeneity in gene expression studies by surrogate variable analysis. *PLoS genetics*, 3(9):e161.

Li, J., Das, K., Fu, G., Li, R., and Wu, R. (2011). The Bayesian lasso for genome-wide association studies. *Bioinformatics*, 27(4):516–523.

Li, Z. and Sillanpää, M. (2012). Estimation of quantitative trait locus effects with epistasis by variational Bayes algorithms. *Genetics*, 190(1):231–249.

Listgarten, J., Kadie, C., Schadt, E. E., and Heckerman, D. (2010). Correction for hidden confounders in the genetic analysis of gene expression. *Proceedings of the National Academy of Sciences*, 107(38):16465–16470.

Logsdon, B., Hoffman, G., and Mezey, J. (2010). A variational Bayes algorithm for fast and accurate multiple locus genome-wide association analysis. *BMC Bioinformatics*, 11(1):58.

Lönnstedt, I. and Speed, T. (2002). Replicated microarray data. *Statistica sinica*, 12(1):31–46.

Luenberger, D. and Ye, Y. (2008). *Linear and Nonlinear Programming*, volume 116 of *International Series in Operations Research & Management Science*. Springer, New York.

Manichaikul, A., Hoffman, E. A., Smolonska, J., Gao, W., Cho, M. H., Baumhauer, H., Budoff, M., Austin, J. H., Washko, G. R., Carr, J. J., et al. (2014). Genome-wide study of percent emphysema on computed tomography in the general population. The Multi-Ethnic Study of Atherosclerosis Lung/SNP Health Association Resource Study. *American journal of respiratory and critical care medicine*, 189(4):408–418.

Marin, J.-M. and Robert, C. P. (2007). *Bayesian Core: A Practical Approach to Computational Bayesian Statistics*. Springer, New York.

Martino, S. and Rue, H. (2009). R package: INLA. *Department of Mathematical Sciences, NTNU, Norway. Available at http://www.r-inla.org*.

McGrory, C. and Titterington, D. (2007). Variational approximations in Bayesian model selection for finite mixture distributions. *Computational Statistics & Data Analysis*, 51(11):5352–5367.

McLachlan, G. and Krishnan, T. (2007). *The EM algorithm and extensions*, volume 382. John Wiley & Sons.

McLachlan, G. and Peel, D. (2004). *Finite Mixture Models*. Wiley, New York.

Michaelson, J. J., Loguercio, S., and Beyer, A. (2009). Detection and interpretation of expression quantitative trait loci (eQTL). *Methods*, 48(3):265–276.

Minka, T. (2001a). Expectation propagation for approximate Bayesian inference. In *Proceedings of the Seventeenth Conference on Uncertainty in Artificial Intelligence*, John Breese and Daphne Koller (Eds.), pages 362–369. Morgan Kaufmann Publishers Inc., San Francisco.

Minka, T. (2001b). *A Family of Algorithms for Approximate Bayesian Inference*. PhD thesis, Massachusetts Institute of Technology.

Nettleton, D. (1999). Convergence properties of the em algorithm in constrained parameter spaces. *Canadian Journal of Statistics*, 27(3):639–648.

Oakes, D. (1999). Direct calculation of the information matrix via the EM. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 61(2):479–482.

Oparina, N. Y., Snezhkina, A., Sadritdinova, A., Veselovskii, V., Dmitriev, A., Senchenko, V., Melnikova, N., Speranskaya, A., Darii, M., Stepanov, O., et al. (2013). Differential expression of genes that encode glycolysis enzymes in kidney and lung cancer in humans. *Russian Journal of Genetics*, 49(7):707–716.

Ormerod, J. (2011). Grid based variational approximations. *Computational Statistics & Data Analysis*, 55(1):45–56.

Ormerod, J. and Wand, M. (2010). Explaining variational approximations. *The American Statistician*, 64(2):140–153.

Price, A. L., Patterson, N. J., Plenge, R. M., Weinblatt, M. E., Shadick, N. A., and Reich, D. (2006). Principal components analysis corrects for stratification in genome-wide association studies. *Nature genetics*, 38(8):904–909.

Risso, D., Ngai, J., Speed, T. P., and Dudoit, S. (2014). Normalization of RNA-seq data using factor analysis of control genes or samples. *Nature biotechnology*, 32(9):896–902.

Rue, H., Martino, S., and Chopin, N. (2009). Approximate Bayesian inference for latent Gaussian models by using integrated nested Laplace approximations. *Journal of the Royal Statistical Society: Series B*, 71(2):319–392.

Saccone, N. L., Schwantes-An, T.-H., Wang, J. C., Grucza, R. A., Breslau, N., Hatsukami, D., Johnson, E. O., Rice, J. P., Goate, A. M., and Bierut, L. J. (2010). Multiple cholinergic nicotinic receptor genes affect nicotine dependence risk in African and European Americans. *Genes, Brain and Behavior*, 9(7):741–750.

Salter-Townshend, M. and Murphy, T. (2009). Variational Bayesian inference for the latent position and cluster model. In *NIPS 2009 (Workshop on Analyzing Networks & Learning with Graphs)*.

Scott-Boyer, M. P., Imholte, G. C., Tayeb, A., Labbe, A., Deschepper, C. F., and Gottardo, R. (2012). An integrated hierarchical Bayesian model for multivariate eQTL mapping. *Statistical applications in genetics and molecular biology*, 11(4).

Shin, S., Park, J. S., Kim, Y.-J., Oh, T., An, S., and Park, C.-S. (2012). Differential gene expression profile in PBMCs from subjects with AERD and ATA: a gene marker for AERD. *Molecular Genetics and Genomics*, 287(5):361–371.

Sing, T., Sander, O., Beerenwinkel, N., and Lengauer, T. (2007). ROCR: Visualizing the performance of scoring classifiers. R package version 1.0-2. `http://rocr.bioinf.mpi-sb.mpg.de/ROCR.pdf/`.

Smídl, V. and Quinn, A. (2005). *The Variational Bayes Method in Signal Processing*. Springer-Verlag, Berlin.

Smyth, G. (2004). Linear models and empirical Bayes methods for assessing differential expression in microarray experiments. *Statistical Applications in Genetics and Molecular Biology*. Volume 3, Issue 1, Article 3.

Smyth, G. (2005). Limma: linear models for microarray data. *Bioinformatics and Computational Biology Solutions using R and Bioconductor*, pages 397–420. Springer, New York.

Stegle, O., Parts, L., Durbin, R., and Winn, J. (2010). A Bayesian framework to account for complex non-genetic factors in gene expression levels greatly increases power in eQTL studies. *PLoS computational biology*, 6(5):e1000770.

Sun, Y., Zhang, N. R., Owen, A. B., et al. (2012). Multiple hypothesis testing adjusted for latent variables, with an application to the AGEMAP gene expression data. *The Annals of Applied Statistics*, 6(4):1664–1688.

Teschendorff, A., Wang, Y., Barbosa-Morais, N., Brenton, J., and Caldas, C. (2005). A variational Bayesian mixture modelling framework for cluster analysis of gene-expression data. *Bioinformatics*, 21(13):3025–3033.

Tzikas, D., Likas, A., and Galatsanos, N. (2008). The variational approximation for Bayesian inference. *Signal Processing Magazine, IEEE*, 25(6):131–146.

Uhl, G. R., Liu, Q.-R., Drgon, T., Johnson, C., Walther, D., Rose, J. E., David, S. P., Niaura, R., and Lerman, C. (2008). Molecular genetics of successful smoking cessation: convergent genome-wide association study results. *Archives of general psychiatry*, 65(6):683–693.

Wand, M. P., Ormerod, J. T., Padoan, S. A., and Frührwirth, R. (2011). Mean field variational Bayes for elaborate distributions. *Bayesian Analysis*, 6(4):1–48.

Wang, B. and Titterington, D. M. (2005). Inadequacy of interval estimates corresponding to variational Bayesian approximations. In *Proceedings of the Tenth International Workshop on Artificial Intelligence and Statistics*, Robert G. Cowell and Zoubin Ghahramani (Eds.), pages 373–380. Society for Artificial Intelligence and Statistics.

Xu, S. (2013). *Principles of statistical genomics*. Springer.

You, M., Wang, D., Liu, P., Vikis, H., James, M., Lu, Y., Wang, Y., Wang, M., Chen, Q., Jia, D., et al. (2009). Fine mapping of chromosome 6q23-25 region in familial lung cancer families reveals RGS17 as a likely candidate gene. *Clinical Cancer Research*, 15(8):2666–2674.

Zhai, C. (2008). Statistical language models for information retrieval. *Synthesis Lectures on Human Language Technologies*, 1(1):1–141.

Zhang, M., Montooth, K., Wells, M., Clark, A., and Zhang, D. (2005). Mapping multiple quantitative trait loci by Bayesian classification. *Genetics*, 169(4):2305–2318.