DEVELOPMENT OF A TISSUE AUGMENTED BAYESIAN MODEL FOR EXPRESSION QUANTITATIVE TRAIT LOCI ANALYSIS

by

YONGHUA ZHUANG

M.D., Tongji University, 2001

Ph.D., Sichuan University, 2009

A thesis submitted to the

Faculty of the Graduate School of the

University of Colorado in partial fulfillment

of the requirements for the degree of

Master of Science

Biostatistics Program

2016

This thesis for the Master of Science degree by

Yonghua Zhuang

has been approved for the

Biostatistics Program

by

Katerina Kechris, Chair and Advisor

Laura M. Saba, Co-Advisor

Stephanie A. Santorico

Date December 16, 2016

Zhuang, Yonghua (M.S., Biostatistics)

Development of a Tissue Augmented Bayesian Model for Expression Quantitative Trait Loci Analysis

Thesis directed by Associate Professor Katerina Kechris and Assistant Professor Laura M. Saba.

ABSTRACT

Expression quantitative trait loci (eQTL) analyses detect genetic variants (SNPs) associated with the RNA expression levels of genes. The conventional eQTL analysis is to perform individual tests for each gene-SNP pair using simple linear regression. The conventional approach to eQTL study is performed on each tissue separately and ignores the extensive information known about RNA expression in other tissue(s). Although Bayesian models have been recently developed to improve eQTL prediction on multiple tissues, they were often based on uninformative priors and/or only evaluated on the number of discoveries in simulated or real data. In this study, we develop a novel tissue augmented Bayesian model for eQTL analysis (TA-eQTL), which takes prior eQTL information from a different tissue into account to better predict eQTL within a tissue. It has been demonstrated that our modified Bayesian model has better performance than several existing methods in terms of sensitivity and specificity using allele-specific expression (ASE) as the gold standard. Compared with other methods, our tissue augmented Bayesian model improves the power and accuracy for cis-eQTL prediction especially when sample size on a tested tissue is small.

The form and content of this abstract are approved. I recommend its publication.

Approved: Katerina Kechris

ACKNOWLEDGEMENTS

I would like to extend my sincerest gratitude to my advisors, Dr. Katerina Kechris and Dr. Laura M. Saba, for their guidance, encouragement, and seemingly limitless patience. Thank you to my committee member, Dr. Stephanie A. Santorico, for sharing her expertise and sage advice. Thank you to Dr. Anna Baron, Dr. Sam MaWhinney, Dr. Gary K. Grunwald, Dr. Edward Bedrick and Dr. Deborah Glueck for their confidence and help. Thank you to Dr. Kenneth L. Tyler and Dr. Penny Clarke for their support.

And most importantly, thank you to my wife, Dan (Dana) Wang, for your love, support, and generosity through all of the late nights and long weekends.

TABLE OF CONTENTS

LIST OF TABLES

TABLE

LIST OF FIGURES

FIGURE

CHAPTER I

INTRODUCTION

I.1 What is eQTL?

Understanding the specific biological effect of genomic variants in cells and tissues provides insight to the biology of disease and complex phenotypes (Nica and Dermitzakis, 2013). Mediating the connection between genetic variants and disease susceptibility may be the RNA expression levels of different genes. Genome-wide association studies (GWAS) have demonstrated that less than 10% of the genetic variants alter coding sequences while more than 90% of genetic variants are located in non-coding regions of the genome for example in promoter regions, enhancers, or even in non-coding RNA genes, which indicates that these genetic variants might be regulatory (Hindorff *et al.*, 2009; Ricaño-Ponce and Wijmenga, 2013; Hrdlickova *et al.*, 2014). The analysis of such genetic variants in the context of gene expression measured in different tissues has established an area of genetics investigating expression quantitative trait loci (eQTL) (Jansen and Nap, 2001).

An eQTL is a locus that explains part of the variation in gene expression levels in either inbred populations (e.g., laboratory mice), or outbred populations (e.g., humans) (Cookson *et al.*, 2009; Nica and Dermitzakis, 2013). An eQTL analysis can help reveal biological processes and discover the genetic factors associated with certain diseases (Nica and Dermitzakis, 2008). Determining if mRNA expression levels are altered by specific genetic variants provides evidence of a mechanistic link between genetic variation and downstream biological events, of which the first step is often changes in gene expression. A standard eQTL study examines the direct association between markers of genetic variation (such as Single Nucleotide Polymorphisms (SNPs)) and mRNA expression levels typically measured in tens or hundreds of individuals. This association can be proximal or distal to the physical location of the gene of interest. The eQTLs that map to the approximate location of the gene are referred to as local eQTL while those that are far from the location of the gene, often on different chromosomes, are referred to as distant eQTLs (Rockman and Kruglyak, 2006). These two types of eQTLs are often referred to as cis and trans, respectively, because local eQTLs are assumed to act in cis and distant eQTLs are assumed to act in trans (Cubillos

et al., 2012). For simplicity, here we use the terms "cis eQTL" for local eQTL and "trans eQTL" for distant eQTL although having a "local" eQTL is not proof of a cis-acting effect (Fraser *et al.*, 2010). Figure 1 illustrates the concept of cis- and trans- eQTL and how they work. Although there is no uniform distance standard to define cis-eQTL, conventionally, variants within 1 Mb (megabase) on either side of a gene's transcription start site (TSS) are considered cis while those variants affecting gene expression at a distance greater than 1 Mb from the TSS or on another chromosome were considered trans-eQTL (Blauwendraat *et al.*, 2016; Webster *et al.*, 2009). Several studies suggest that most of the regulatory control takes place locally, in the vicinity of genes (Dixon *et al.*, 2007; Göring *et al.*, 2007; Schadt *et al.*, 2008). Numerous genes have been detected to have cis-eQTLs while detecting trans-eQTLs has been more challenging. Of note, some cis-eQTLs are detected in many tissue types while the majority of trans-eQTLs are tissue-dependent (Gerrits *et al.*, 2009).

I.2 Recombinant inbred (RI) mouse strains

Recombinant inbred (RI) strains have been used widely for quantitative trait mapping and are favorable for eQTL studies (Pandey and Williams, 2014). RI strains are fully inbred strains that are produced by intercrossing two parental strains and followed by repeated sibling matings for at least 20 generations. Each RI strain characterizes a unique and fixed chromosomal mosaic of the parental genomes (Pandey and Williams, 2014). The most important advantage of RI strains is that phenotypes and eQTL studies can be combined to construct phenome-genome association because the genetic background of a strain is held constant over generations.

The BXD RI set, the largest and oldest RI family, was generated by crossing C57BL/6J (B) females with DBA/2J (D) males. The BXDs have been used to study complex traits since the mid 1970s and the genetics of gene expression since the early 2000s (Pandey and Williams, 2014). In addition to the remarkably deep phenome data sets available for the BXDs, a further advantage is that both parents have been sequenced completely (Keane *et al.*, 2011; Carneiro *et al.*, 2009). A complete compendium of C57BL/6J (B) versus DBA/2J (D) sequence variants is available online (www.genenetwork.org) and can be used to identify causal SNPs. In addition, it is feasible to use reverse genetic methods with

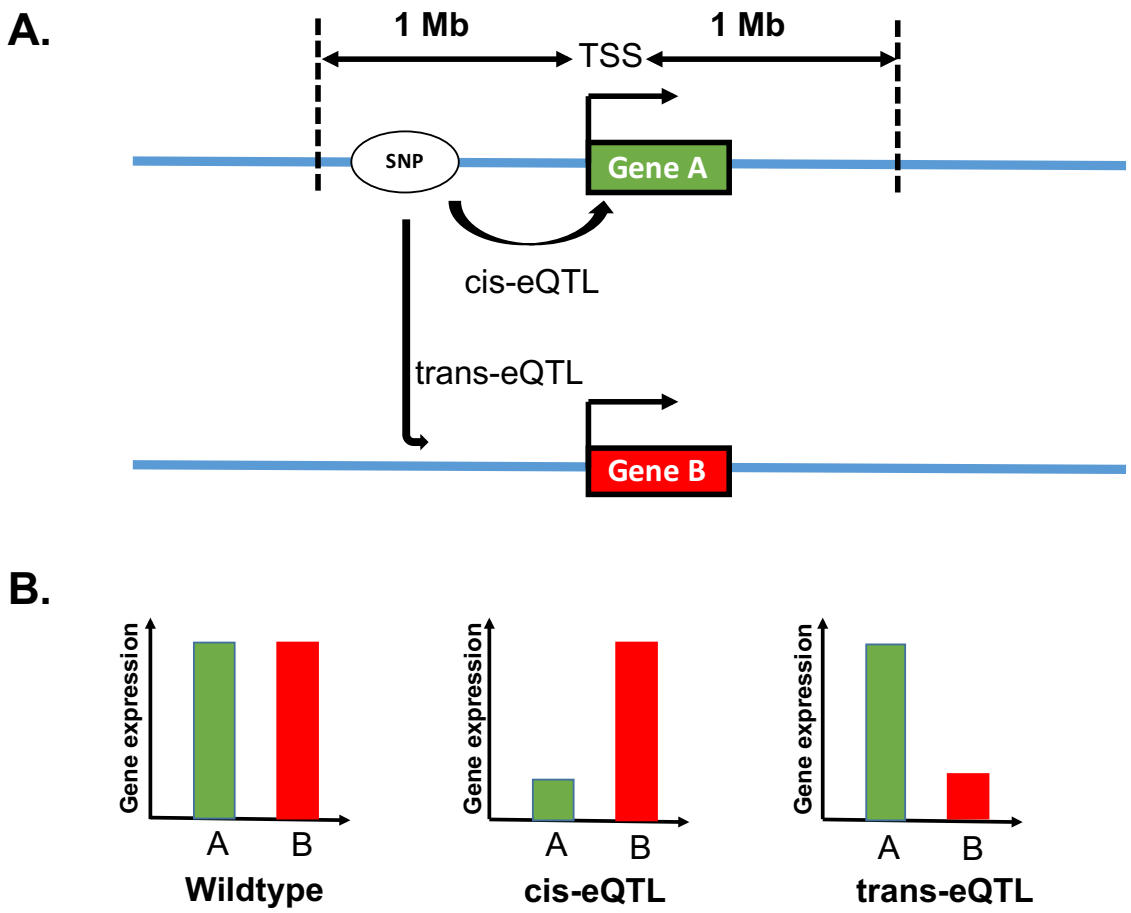

Figure I.1: Illustration of cis and trans expression quantitative trait loci (eQTLs). (**A**) SNP, white circle; gene A, green rectangle (same chromosome); gene B, red rectangle (different chromosome). Each blue line represents different chromosomes. (**B**) in wild-type, gene A (green bar) and gene B (red bar) are highly expressed in wild-type. In cis-eQTL, the gene A expression (green bar) is inhibited due to the SNP on the same chromosome while the transcription level of gene B is not changed. In trans-eQTL, the expression of gene B (red bar) is down-regulated by SNP on the other chromosome.

the BXDs to track down those phenotypes that map to the location of a particular sequence variant. The current BXD panel contains around 120 lines that are almost fully inbred and available from the Jackson Laboratory, and another set of 30-40 that are being inbred by Williams, Lu, and his colleagues at University of Tennessee Health Science Center. Of note, the BXD family of RI strains —has around 100 independent eQTL studies at the end of 2014, all of which has been assembled at the GeneNetwork web site (www.genenetwork.org) (Pandey and Williams, 2014).

I.3 Allele-specific expression (ASE)

A powerful approach for identifying cis-eQTL is measuring allele-specific expression (ASE) in a diploid. ASE describes the situation where the two alleles of a gene are expressed at different levels. An observation of differential allelic gene expression in a heterozygote indicates that one or more variants have arisen and acted in cis to affect the expression level of the gene (Skelly *et al.*, 2011). ASE has been studied by a variety of methods, including allele-specific PCR (Ronald *et al.*, 2005), pyrosequencing (Wittkopp *et al.*, 2004), allele-specific gene expression arrays (Serre *et al.*, 2008) and next generation RNA sequencing (Skelly *et al.*, 2011).

I.4 Current methods for eQTL analysis

The conventional approach to eQTL analysis is to perform individual tests for each gene-SNP pair using simple linear regression with the number of minor alleles as the predictor variable. Figure 2 depicts the typical analysis strategy for single gene-SNP association. To choose the most promising SNPs for further evaluation and analysis, the traditional approach simply selects the SNPs with the smallest association P values from standard maximum likelihood tests (Chen and Witte, 2007).

I.5 Challenges and limitations of conventional methods

This conventional method for eQTL study suffers several limitations. The eQTL analysis with linear regression assumes that every SNP has an equal likelihood of causality and works independently on the targeted gene, which might not be the case. In the conventional study, the huge number of genetic markers and expression traits and their complicated correlations lead to a multiple-testing problem (Zhang *et al.*, 2012). How to appropriately make corrections for multiple-testing is challenging for eQTL studies. In addition, causal SNPs may not exist or be genotyped for some targeted genes. The conventional eQTL linear regression is performed on each tissue separately and ignores the extensive information known about the SNPs effect on RNA expression in the other tissue(s), which results in low power and less accuracy due to a limited sample size in the tissue of interest. To solve these problems, several approaches including Bayesian modeling have been developed.

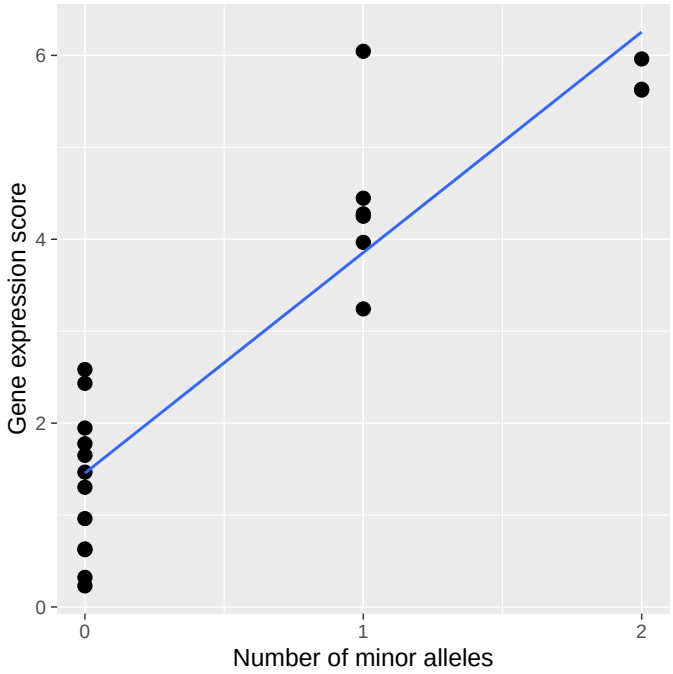

Figure I.2: eQTL analysis with a simple linear regression model. Each black dot represents individual gene expression estimates with corresponding number of minor alleles for an individual. The blue line is the best-fit line derived from simple linear regression using least square estimates.

I.6 Current Bayesian models and limitations

Bayesian prediction is a method of statistical inference in which Bayes' theorem is used to update the probability for a hypothesis when more information becomes available. Bayesian models have been introduced for eQTL and GWAS studies (Scott-Boyer *et al.*, 2012; Veyrieras *et al.*, 2008; Stegle *et al.*, 2010; Stephens and Balding, 2009; Chen and Witte, 2007). Bayesian methods provide a natural modeling framework for eQTL analysis, where information shared across markers and/or genes can increase the power to detect eQTLs (Chen and Witte, 2007; Imholte *et al.*, 2013). Bayesian models are usually based on some modification of a linear model relating expression to SNP genotype(s) (Veyrieras *et al.*, 2008; Stegle *et al.*, 2010; Chen and Witte, 2007). In most cases, non-informative priors are assigned or hyper-parameters for the priors are set to arbitrary values. To date, most eQTL analyses have studied the association of gene and SNP within a single tissue. In only a few studies, the informative priors of eQTL include information on that eQTL from a different tissue (Li *et al.*, 2016; Flutre *et al.*, 2013). Recently, Dr. Li and colleagues developed an empirical

Bayes approach for multiple tissue eQTL analysis (MT-eQTL) (Li *et al.*, 2016). Although MT-eQTL accommodates variation in the number of samples, it was not designed to deal with the unequal number of gene transcripts among multiple tissues. In other words, MT-eQTL method only performs analysis on the overlapping gene probesets for eQTL prediction and ignores other transcript information which are not in all tested tissues.

In terms of model performance evaluation, to our knowledge, current Bayesian models have been evaluated on the power for detecting associated SNPs either on simulated data or based on the number of discoveries on the real data. Performance assessment on real data is often limited because of an overemphasis on the number of detected eQTLs while ignoring potential false positives. The performance of prediction models can be better assessed using other methods and benchmarks, such as allele-specific expression (ASE).

I.7 Hypothesis and goals

At the molecular level, comparisons across tissues and species are often conducted to identify conserved expression changes. For eQTL, we hypothesize that mechanisms for transcriptional control through SNPs may be conserved across tissues and species and integrating known eQTL results in one tissue/species to inform the prediction of eQTL in another tissue/species will improve power and accuracy. We can establish this approach for cross-tissue or even cross-species studies and study genomic variants and their impact on gene expression.

Specifically, in the following study, we hypothesize that integrating mouse lung eQTL results will help inform the prediction of mouse liver eQTL. Since eQTL analysis, especially trans-eQTL detection, is a computationally intensive task, we focus on cis-eQTL analyses in this study and develop a tissue augmented Bayesian model of eQTL (TA-eQTL) to improve the accuracy of cis-eQTL prediction. In the future, we will extend our study and optimize the augmented Bayesian model we develop here to improve human eQTL prediction by incorporating mouse eQTL information.

I.8 Novelty

In this study, we incorporate results of mouse lung eQTL (RI mouse panel) to increase

power and accuracy of liver eQTL prediction. We develop a novel Bayesian model for eQTL analysis, which takes prior eQTL information into account to better predict eQTL in another tissue. This novel model could also be applied to improve human cis-eQTL prediction by incorporating another species (such as mouse) information. The current Bayesian models were often evaluated based on the simulated data (Li *et al.*, 2016; Das *et al.*, 2015; Flutre *et al.*, 2013) or only on a small number of previously known causal SNPs (Chen and Witte, 2007). In this study we first evaluate model performance with several methods based on liver ASE-verified cis-eQTL data rather than only utilizing simulated data. We also assess model performance by sub-sampling.

CHAPTER II

DATA

II.1 Study subjects: BXD inbred mice

Gene expression data and SNP genotypes in BXD inbred mice were downloaded from the GeneNetwork website (Chesler *et al.*, 2004; Wang *et al.*, 2003). The BXD family of recombinant inbred (RI) strains were derived by crossing C57BL/6J (B6) and DBA/2J (D2) inbred mouse strains and inbreeding progeny for 20 or more generations. The BXD RI strains has been successfully used to study the genetics of several behavioral phenotypes including alcohol and drug addiction, stress, and locomotor activity (Tabakoff *et al.*, 2008; Phillips *et al.*, 1995).

II.2 Liver gene expression data

The liver gene expression data for BXD inbred mice from GEO series GSE16780 were downloaded from the GeneNetwork website. These data were generated by Dr. Jake Lusis and colleagues at UCLA using GeneChip® Mouse Genome 430A Array and are currently listed as a BXD data set, although the study includes many other strains included in the Hybrid Rat Diversity Panel (Bennett *et al.*, 2010). The GeneChip® Mouse Genome 430A Array from Affymetrix is a single array representing approximately 14,000 well-characterized mouse genes that can be used to explore biology and disease processes.

RNA was isolated from livers of 99 mouse strains including 30 BXD stains. Double stranded cDNAs were synthesized from 1 μg total RNA through reverse transcription with an oligodT primer using the cDNA Synthesis System. Biotin−labeled cRNA was generated from the cDNA and hybridized to Affymetrix Mouse Genome 430A arrays (20634 probe-sets). Array hybridization, washing and scanning were performed using the manufacturer's protocol. The scanned image data was processed using the Affymetrix GCOS software and the Robust MultiArray method (RMA) to estimate the RNA expression levels of each gene (Bennett *et al.*, 2010). Expression of transcripts in the liver as well as most other GeneNetwork data sets is reported on a log2 scale. In other words, a one unit difference corresponds approximately to a 1-fold difference (doubling of expression) in hybridization signal inten-

sity. In order to simplify comparisons among different data sets, log2 RMA values of each array were adjusted to an average expression of 8 units and a standard deviation of 2 units (variance stabilized).

II.3 Lung gene expression data

The lung gene expression data set for 47 BXD strains of mice were generated using the Mouse Genome 430 2.0 Affymetrix array and normalized expression data were downloaded from GeneNetwork website. The Affymetrix Mouse Genome 430 2.0 Array offers complete coverage of the Mouse Expression Set 430 and 430A for analysis of over 39,000 transcripts on a single array. The data set includes 47 BXD strains and reciprocal F1 hybrids (B6D2F1 and D2B6F1). Data were generated by Klaus Schughart, Lu Lu, and Rob Williams. Arrays were processed using RMA protocol by Yan Jiao and Weikuan Gu at the Memphis Veteran Affairs Medical Center (VA)(Alberts *et al.*, 2011).

RNA was isolated from 47 strains of the BXD RI panel. Double-stranded cDNAs were synthesized from 8 μg total RNA using a standard Eberwine T7 polymerase method. The Affymetrix IVT labeling kit (Affy 900449) was used to generate labeled cRNA. 4-5 μg of each biotinylated cRNA preparation was fragmented and hybridized for 16 hours. After hybridization, GeneChips were washed, stained with streptavidin-phycoerythrin (SAPE), and read using an Affymetrix GeneChip fluidic station and scanner according to the manufacturer's protocol (Alberts *et al.*, 2011). Expression of transcripts in the lung is also reported on a log2 scale.

Of note, the gene expression from GeneNetwork in both liver and lung tissues includes 30 and 47 strains of BXD inbred mice, respectively. The 30 strains of BXD mice in the liver gene expression dataset are all included the 47 strains of lung dataset. Of note, only 45 BXD strains in lung expression data have SNP information available.

II.4 Genotype data (SNP) for BXD

The BXD genotype data file were downloaded from GeneNetwork website (http://www.genenetwork.org/genotypes/BXD.geno) on November 30, 2015. A total of 96 BXD strains with 3811 SNPs were obtained. The great majority of SNP genotypes were generated on

the Illumina SNP BeadArray. The heterozygous SNPs were excluded from analysis due to their uncertainty of their validity.

II.5 Allele-specific expression (ASE) in mouse liver

Dr. Lagarrigue and colleagues have analyzed allele-specific expression (ASE) and parent-of-origin expression in adult mouse liver using next generation sequencing (RNA-Seq) in reciprocal crosses of heterozygous F1 mice from the BXD RI parental strains, C57BL/6J and DBA/2J (Lagarrigue *et al.*, 2013). In this study, they utilized a 10-Mb window on either side of the gene for the classification of local eQTL. An exon was considered to have ASE if P-value ≤ 0.05 and the B/D expression ratio is significantly greater than 1.5 or less than 1/1.5 . P values were calculated using a Fisher's exact test with the Benjamini-Hochberg method adjustment to control multiple testing. Dr. Lagarrigue and colleagues found, in three diet and sex contexts, 397 exons (284 genes) under ASE and shared by two replicates. They reported that a 60% overlap between genes exhibiting ASE and putative cis-eQTL identified in an intercross between the same strains. Among the 284 ASE genes that replicated among samples, 170 (60%) overlap with the 2382 local-eQTL genes published in a previous study by Dr. Lagarrigue (Davis *et al.*, 2012). Of note, 272 ASE genes were found in mice with mouse standard diet (chow).

We downloaded all significant ASEs identified in chow-fed mice from http://www.genetics.org and used them as a "gold standard" to evaluate the performance of our newly developed Bayesian methods. In other words, only these 272 ASE genes are considered to have true cis-eQTL while the others are not considered to have a true cis-eQTL.

CHAPTER III

METHODS

In this study, unless otherwise specified, all data manipulation and data analyses were performed using RStudio (version 0.98.1091)(RStudio Team, 2015), R (version 3.2.3)(R Core Team, 2015) using the following packages: "MatrixEQTL"(Shabalin, 2012), "ggplot2"(Wickham, 2009), "fBasics"(Team *et al.*, 2014), "xtable"(Dahl, 2016), "biomaRt"(Durinck *et al.*, 2005), "plyr" (Wickham, 2011), "data.table" (Dowle *et al.*, 2015), "flux" (Jurasinski *et al.*, 2014) , "pROC" (Robin *et al.*, 2011), "lme4" (Bates *et al.*, 2015) and "lsmeans" (Lenth, 2016).

III.1 SNP data pre-processing

The original SNP data include 3811 markers on 93 BXD stains mice. These SNPs are located on Chromosomes 1-19 and Chromosome X. The SNPs in BXD inbred mice were originally coded as "B", "D", "H" (heterozygous) and "U" (unknown). They were recoded as "0", "1", "NA" and "NA", respectively. In other words, heterozygous genotypes and unknown genotypes were set to missing. The SNP locations were updated to the Ensembl variation 85: Mus musculus genes (GRCm38.p4) version using Biomart online tool (http://uswest.ensembl.org/biomart/). Among 3811 SNP markers, the chromosome locations were only available for 3023 SNPs in the GRCm38.p4 annotation database.

III.2 RNA expression data pre-processing

The gene expression data in mouse liver and lung obtained from Mouse Genome 430A Array (22690 probesets) and 430 V2 Arrays (45119 probesets) were annotated with Ensembl 85: Mus musculus genes (GRCm38.p4) using the Biomart online tool (http://uswest.ensembl.org/biomart/) to retrieve the transcript corresponding gene Ensembl gene ID and gene location. Of note, we were only able to retrieve Ensembl gene IDs and gene locations for 20651 probesets(corresponding to 12736 unique genes) and 33684 probesets (corresponding to 12736 unique genes) probe sets in liver and lung expression data, respectively.

III.3 Basic Bayesian cis-eQTL analysis

Prior to developing Bayesian models, we examined whether the shared cis-eQTLs be-

tween mouse lung and mouse liver are significantly overrepresented at different thresholds of P value using a Chisq test. A p value < 0.05 is considered significant.

We extended the basic Bayesian linear regression framework (Chen and Witte, 2007; Gelman *et al.*, 2014) and developed a model that does not assume non-informative or arbitrary priors. To set informative priors, we analyzed lung cis-eQTL on a panel of recombinant inbred mice (45 strains with available SNP information).

To get prior information from mouse lung tissue, a model for eQTL analysis is

$$y_{lgi} = \alpha_{lgk} + \beta_{lgk} x_{ki} + \varepsilon_{lgki}, \tag{III.1}$$

- y_{lgi} is the mean expression level of gene g in strain i and lung (l) tissue;
- α_{lgk} is the tissue (l, lung), gene (g), and SNP (k) specific intercept;
- β_{lgk} is the tissue (l, lung), gene (g), and SNP(k) specific coefficient;
- x_{ki} is the genotype for SNP k and strain i coded as 0 and 1;
- ε_{lgki} is the error term for strain i, gene g, tissue l, and SNP k;

The error term is assumed to have a Gaussian distribution, $N(0, \sigma^2_{lgk})$. Each cis-SNP is modeled and regressed separately against each gene. In both the liver and lung BXD studies, the environmental and genetic parameters were tightly controlled. Thus, no additional covariates were adjusted.

As with the mouse lung eQTL analysis, a similar basic model relating liver gene expression to genotype is

$$y_{vgi} = \alpha_{vgk} + \beta_{vgk} x_{ki} + \varepsilon_{vgki}, \tag{III.2}$$

- y_{vgi} is the mean expression level of gene g in the strain i and the tissue liver (v);
- α_{vgk} is the tissue (v, liver), gene (g), and SNP (k) specific intercept;
- β_{vgk} is the tissue (v, liver), gene (g), and SNP(k) specific coefficient;
- x_{ki} is the genotype for SNP k and strain i coded as 0 and 1;

- ε_{vgki} is the error term for strain i, gene g, tissue v (liver), and SNP k;

The error term is assumed to have a Gaussian distribution, $N(0, \sigma^2_{vgk})$. Each cis-SNP is modeled and regressed separately against each gene. As in lung, no additional covariates were adjusted.

For simplicity, we only select the gene-SNP pair with minimum P value at each gene level (β_{lgm} and β_{vgm}) for Bayesian prediction. In other words, each gene has only one cis-eQTL for further analysis. The specific SNP selected to represent the cis-eQTL for each gene might not be the same between liver and lung tissues.

- β_{lgm} is the specific coefficient for gene (g) and SNP with minimum P value (m) in tissue (l, lung);
- β_{vgm} is the specific coefficient for gene (g) and SNP with minimum P value (m) in tissue (v, liver);

The parameter of interest, the regression coefficient for mouse liver, can be first estimated using the basic model (no prior) with the Matrix eQTL package (Shabalin, 2012). In this study, we ignored the directionality of β since the direction of the effect in mouse lung is not relevant to mouse liver because we do not force the exact same genetic variant to be used in both tissues. Thus, we took the absolute values of β_{lgm} and β_{vgm} for further analyses. Both $|\beta|_{lgm}$ and $|\beta|_{vgm}$ represent the effect size of SNPs on gene expression. For simplicity, we drop the m subscript for $|\beta|_{lgm}$ and $|\beta|_{vgm}$. To further inform the estimation of $|\beta|_{vg}$ for mouse liver genes using additional prior information, we assume,

$$|\beta_{vg}| = Z_{lg}\Gamma + U_g, \quad U \sim \mathcal{N}(0, \tau^2) \tag{III.3}$$

- $|\beta_{vg}|$ is a vector of the absolute first-stage coefficients (III.2) for the gene (g) and SNP pair with minimum P value in liver (v) tissue;

$$\underset{g\times 1}{|\beta_{vg}|} = \begin{bmatrix} |\beta_{v1}| \\ |\beta_{v2}| \\ |\beta_{v3}| \\ \dots \\ |\beta_{vg}| \end{bmatrix}$$

- Z_{lg} is a vector including the intercept and the absolute first-stage coefficients (III.2) for the gene (g) and SNP pair with minimum P value in lung (l) tissue;

$$\underset{g\times 2}{Z_{lg}} = \begin{bmatrix} 1 & |\hat{\beta}_{l1}| \\ 1 & |\hat{\beta}_{l2}| \\ 1 & |\hat{\beta}_{l3}| \\ \dots & \dots \\ 1 & |\hat{\beta}_{lg}| \end{bmatrix}$$

- Γ is a coefficient vector corresponding to the additive contribution of the features to the prior mean;

$$\underset{2\times 1}{\Gamma} = \begin{bmatrix} \gamma_1 \\ \gamma_2 \end{bmatrix}$$

- U is the error matrix with zero mean and variance τ^2.

The prior features we considered for inclusion are the significance level (negative logarithm of p-value) and effect magnitude of each mouse SNP and gene association (absolute value of estimated β_{lg}).

The Gaussian conjugate prior assumption leads to a closed form solution to estimate β that simplifies computation. By completing the square, for one gene the posterior distribution of β, given the data, is Gaussian with posterior mean,

$$\tilde{\beta} = (1-\lambda)Z\hat{\Gamma} + \lambda\left|\hat{\beta}\right| \tag{III.4}$$

which is the weighted average of the maximum likelihood estimate (MLE) $\hat{\beta}$ using the basic model (no prior) and the prior mean $Z\Gamma$ (Chen and Witte, 2007).

The "shrinkage" term λ is a function of the two variances, σ^2_{vgk} from the basic model (III.2) and τ^2 from the prior in the second stage model. λ indicates how much the MLE is shrunk towards the prior mean $Z\hat{\Gamma}$. λ increases to 1 when τ^2 is large (e.g., less informative prior of mouse lung eQTL) and σ^2_{vgk} is small, therefore giving less influence on prior, while λ decreases to 0 when τ^2 is small (more informative prior) and σ^2_{vgk} is large, thereby giving more influence to the prior. Least squares is used in the basic model to obtain estimates

$\hat{\beta}$ and σ^2_{vgk}. For estimating $\hat{\gamma}$ and τ^2, a least squares method can also be employed. We assume a common variance and independence across all SNPs and start modeling with an identity matrix. These estimates are substituted into the shrinkage term and the expression for the posterior mean.

III.3.1 Weighted Bayesian model

In a standard Bayesian model, we found that estimates in the second stage model ($Z\hat{\Gamma}$) are much less than their corresponding $\left|\hat{\beta}\right|$, based on the basic model without prior knowledge. Thus, we introduced a constant (c) weight to rescale the final estimate $\tilde{\beta}$.

$$c = \frac{max(\left|\hat{\beta}\right|)}{max(Z\hat{\Gamma})} \tag{III.5}$$

The weighted Bayesian posterior mean is estimated by,

$$\tilde{\beta} = c(1-\lambda)Z\hat{\Gamma} + \lambda\left|\hat{\beta}\right| \tag{III.6}$$

III.3.2 Variance of posterior mean and posterior probability

The conjugate prior for liver β was assumed to have a normal distribution: $|\beta_{vg}| \sim \mathcal{N}(Z\Gamma, \tau^2)$. The posterior distribution for $|\beta_{vg}|$ is (Kulis, 2012):

$$|\beta_{vg}| \left|\beta_{lg}, \tau^2, \sigma^2_{vg} \sim \mathcal{N}(\tilde{\beta}, S), \right. \tag{III.7}$$

where

$$S^{-1} = (\tau^2)^{-1} + (\sigma^2_{vg})^{-1} \tag{III.8}$$

After calculating the posterior mean $\tilde{\beta}$ and variance S, we determined the posterior probability of $\tilde{\beta}$ below 0, $\mathrm{P}(\tilde{\beta} < 0|\beta_{lg}, \tau^2, \sigma^2_{vg})$, using the "pnorm" function in R (version 3.2.3).

III.4 Model performance evaluation

Compared with simulation studies in pre-existing methods, we evaluated our method

with both existing and novel strategies. We named the weighted Bayesian model we developed in this study the tissue augmented Bayesian model of eQTL (TA-eQTL).

III.4.1 Models evaluation based on ASE

To evaluate TA-eQTL method, we compared the results with liver cis-eQTL benchmarks verified in allele specific expression studies. We used the significant ASEs (Lagarrigue *et al.*, 2013) as a standard to evaluate the performance of newly developed Bayesian methods. Only these 272 ASE genes are considered to have true liver cis-eQTL while all other mouse genes were not considered to have liver cis-eQTL. According to the ASE gold standard, we were able to determine the sensitivity and specificity of testing methods, which enables us to derive Receiver operating characteristic (ROC) curves and compare the power and accuracy between Bayesian models and other existing approaches. The area under the ROC curve was computed following the trapezoid rule and the 95% confidence interval (CI) was determined with 2000 stratified bootstrap replicates (Robin *et al.*, 2011). The DeLong's significance test (DeLong *et al.*, 1988) was performed to compare the AUCs of two correlated ROC curves with the "roc.test" function in "pROC" package (Robin *et al.*, 2011).

III.4.2 Comparison with other methods

We compared the performance of TA-eQTL method with other existing methods, such as the conventional model (linear regression in liver dataset without lung prior information), meta-analysis approach (Stouffer S, 1949; T., 1958), and an empirical Bayes approach for multiple tissue eQTL analysis (MT-eQTL) (Li *et al.*, 2016). We also added the linear regression result on lung only, which served as a control.

For meta-analysis, we used the Stouffer test. Stouffer's method converts one-tailed P values (P_i) from each of k independent tests into standard normal deviates (Z_i) and determines the Z_S score ($Z_S = \frac{\sum_{i=1}^{k} Z_i}{\sqrt{k}}$) to estimate an overall p value "(Stouffer S, 1949). Stouffer's method is known as the "inverse normal" or "Z-tranform" method (Stouffer S, 1949; Whitlock, 2005). Of note, Liptak advanced Stouffer's method by assigning different weights (W_i) to each study, $Z_w = \frac{\sum_{i=1}^{k} W_i Z_i}{\sqrt{\sum_{i=1}^{k} W_i^2}}$. When each test has equal weighting, this reduces to the Stouffer test procedure. Liptak's method is known as the "Liptak-Stouffer"

or "weighted Z-transform" method (Laoutidis and Luckhaus, 2015). We used two-sized P value test for the conventional liver and lung cis-QTL analyses (not a one-sided P value). The two-sided test is appropriate because we are interested in $|\beta|$, but not the directionality of β.

In the MT-eQTL method, a hierarchical Bayesian model for a vector Z_λ of Fisher transformed correlations between expression and genotype across tissues is assumed (Li *et al.*, 2016), where $Z_\lambda|\mu_\lambda \sim \mathcal{N}_k(\mu_\lambda, \Delta)$ and μ_λ denotes the true effect sizes of the gene-SNP pair λ across the k tissues. The covariance matrix Δ has diagonal values 1 and its off-diagonal values capture the correlations between tissues. In MT-eQTL estimation, $\mu_\lambda = \Gamma_\lambda \alpha_\lambda$, where Γ_λ and α_λ are two random vectors. The prior Γ_λ indicates whether there is an eQTL in each of the k tissues and α_λ is a effect size vector for the gene-SNP pair λ. The marginal posterior probability of having an eQTL in each tissue is $P(\Gamma_{\lambda k} = 1|Z_\lambda)$. The MT analysis reports the marginal probability of not having an eQTL, $P(\Gamma_{\lambda k} = 0|Z_\lambda)$, in each tissue. Smaller values of the marginal probability of not having an eQTL indicate higher likelihood of the gene-SNP pair being an eQTL in the tissue (Li *et al.*, 2016). In our present study, we focused on detecting the cis-eQTL at a gene level. Thus, we selected the gene-SNP pair with minimum marginal probability of not having an eQTL on liver tissue at gene level for model performance comparison.

III.4.3 Model evaluation by sub-sampling

We hypothesized that the new augmented Bayesian model improves the power and accuracy for cis-eQTL prediction when the sample size is small. When sample size decreases, prior information may help and make up for small sample size. To address the effect of sample size in our newly developed TA-eQTL method, we sub-sampled the strains in the liver gene dataset but maintained the prior information from the complete lung eQTL data analysis. The conventional liver gene expression data includes 30 BXD strains and we randomly subsetted them to 10 strains, 15 stains, 20 strains and 25 strains without replacement. Each subsetting was performed six times with random samplings. For each sampling, we calculated the P value in the conventional liver cis-eQTL analysis, the posterior probability below 0 in the TA-eQTL prediction, the marginal P values in the multiple tissue (MT) anal-

ysis, the P value from the meta-analysis and the P values in the conventional lung analysis. Then, the mean values of these P values or probability values were used to derive ROC curves and compare model performance. In addition, we also calculated the AUC among the five tested methods at each random sampling (6 samplings) and subsetting (4 subsettings). Linear mixed models were then used to compare the cis-eQTL analyzing methods. These models accounted for random effect of sub-sampling and the correlation of samples. The regressions were performed using the "lmer" function in "lme4" package (Bates *et al.*, 2015) and the pair comparisons were done with the "lsmeans" function in "lsmeans" package (Lenth, 2016).

CHAPTER IV

RESULTS

IV.1 Overlap of lung and liver cis-eQTL

We performed cis-eQTL analysis on 20651 (corresponding to 12736 unique genes) and 33684 (corresponding to 12736 unique Ensembl annotated genes) probe sets with 3023 SNPs for mouse liver and lung, respectively. 10579 genes were found to have potential cis-eQTL, i.e., there is one or more SNPs within 1 Mb on either side of their transcription start site (TSS). From the potential cis-eQTL for a gene, we selected the SNP with the minimum P value in each tissue for further analyses. First, we examined whether local genomic control of transcript expression levels is conserved across tissues in mice by comparing the observed overlap of conventionally calculated cis-eQTL with the expected overlap between liver and lung. The expected number of shared cis-eQTL was calculated under the assumption that the likelihood of a cis-eQTL in the two tissues is independent.

We found that the observed number of shared cis-eQTL between liver and lung is significantly higher ($P < 0.05$) than the expected overlap at several different P value thresholds for declaring a cis-eQTL significant (Figure IV.1, Supplemental Table A.1). We also observed that the ratio of observed vs. expected (ratio=$\frac{Observed\ \ shared\ \ cis\ \ eQTL}{Expected\ \ shared\ \ cis\ \ eQTL}$) is positively associated with negative log P value (Figure IV.1). The ratio is 1.71 when the cis-eQTL P value threshold is 0.05. The ratio increases to 9.06 as the cis-eQTL P-value threshold becomes more stringent, i.e., negative log P values increases (Figure IV.1, Supplemental Table A.1). We also summarized the number of significant cis-eQTL at different P values thresholds in lung and liver and found that although lung has more, the two sets are fairly similar (Table IV.1). All of the above suggests that the mechanisms for gene expression control through local SNPs is conserved across tissues, i.e., different tissues share cis-eQTL. Thus, it may be useful to take advantage of the known cis-eQTL information in one tissue to help predict unknown cis-eQTL in another tissue.

IV.2 Unweighted Bayesian model

Next, we developed an augmented Bayesian modeling approach to identify liver cis-

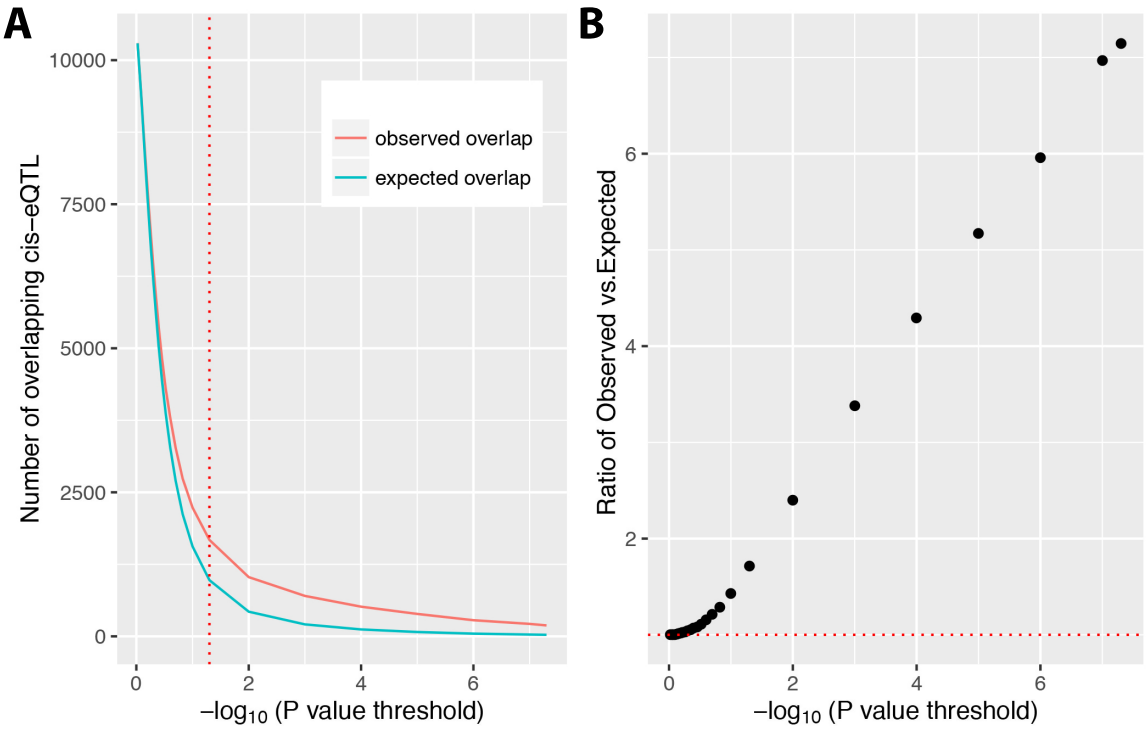

Figure IV.1: Comparison of overlapping cis-eQTL between mouse liver and lung. (A) Number of observed and expected overlapping liver and lung cis-eQTL at different cis-eQTL significance thresholds. The red dotted line is a nominal threshold of P value 0.05 ($-log_{10}(0.05) = 1.3$). The cyan curve represents the expected number of overlapping cis-eQTL, which is calculated under the assumption that the probability of a significant cis-eQTLs in mouse liver and lung are independent. The red curve represents the observed number of overlapping cis-eQTL. (B) The ratio of the number of observed overlapping cis-eQTL to the number of expected overlapping cis-eQTL at different significance thresholds. Ratio=$\frac{Observed\ \ shared\ \ cis\ \ eQTL}{Expected\ \ shared\ \ cis\ \ eQTL}$. Each black dot represents the ratio between observed and expected cis-eQTL in two tissues at different cis-eQTL P value thresholds. The red dotted line represents $ratio = 1$.

eQTL using lung cis-eQTL results in mice as prior information. To get informative priors, we identified lung cis-eQTL in the BXD RI panel using a standard linear regression approach. As with the lung cis-eQTL analysis, we also performed liver cis-eQTL analysis using simple linear regression. Then, we incorporated lung cis-eQTL information including β and/or negative log P values into the Bayesian model as prior information to enhance liver cis-eQTL prediction. Since there is significant correlation between β magnitude and negative log P values($\rho = 0.84, P < 2.2e-16$) (Figure IV.2), we only chose one of them to include

Table IV.1: Summary of genes with a significant cis-eQTL in each tissue with different cis-eQTL P value threshold using the conventional method.

P value threshold	No. of genes with a significant cis-eQTL in lung (% of total)	No. of genes with a significant cis-eQTL in liver (% of total)
0.05	3609 (34)	2858 (27)
0.01	2477 (23)	1828 (17)
0.001	1774 (17)	1238 (12)
1e-04	1381 (13)	919 (9)
1e-05	1124 (11)	708 (7)
1e-06	922 (9)	539 (5)
1e-07	777 (7)	422 (4)
1e-08	665 (6)	315 (3)
1e-09	550 (5)	245 (2)

as prior information. Since β magnitude in lung cis-eQTL has a similar range to the β magnitude in liver cis-eQTL, we used $|\beta|$ in the lung cis-eQTL as a prior for Bayesian model development. We also tried the negative log of P values as prior for Bayesian model instead of $|\beta|$ in the lung cis-eQTL, but the results do not differ much in these two strategies for choosing prior information (data not shown).

Next, we used a standard Bayesian model (unweighted) to incorporate lung cis-eQTL information to update the liver results. We found that in unweighted Bayesian analysis, posterior estimates ($\tilde{\beta}$) were generally lower than the conventional liver prediction ($\left|\hat{\beta}\right|$) (Figure IV.3). This is because the prior mean of cis-eQTL ($Z\hat{\Gamma}$) are lower than the conventional liver estimates ($\left|\hat{\beta}\right|$). The maximum of the prior estimates of the cis-eQTL effect in liver, $Z\hat{\Gamma}$, is 1.17, while the maximum of the estimate of the cis-eQTL effect in liver derived directly from the liver data, $\left|\hat{\beta}\right|$, is 5.18 (Supplemental Table A.2).

IV.3 Weighted Bayesian model

To adjust for the distribution difference between $Z\hat{\Gamma}$ and $\left|\hat{\beta}\right|$, we introduced a weight to the Bayesian model. We calculated the weight based on the maximum values of $\left|\hat{\beta}\right|$ and $Z\hat{\Gamma}$, $c = \frac{max(|\hat{\beta}|)}{max(Z\hat{\Gamma})}$.

As shown in Figure IV.4, the weighted Bayesian model corrects the imbalance between $Z\hat{\Gamma}$ and $\left|\hat{\beta}\right|$ to influence the posterior estimates ($\tilde{\beta}$). Next, we calculated the variance of the posterior distribution based on σ^2_{vg} and τ^2 (See Chapter III). To rank the liver cis-eQTL

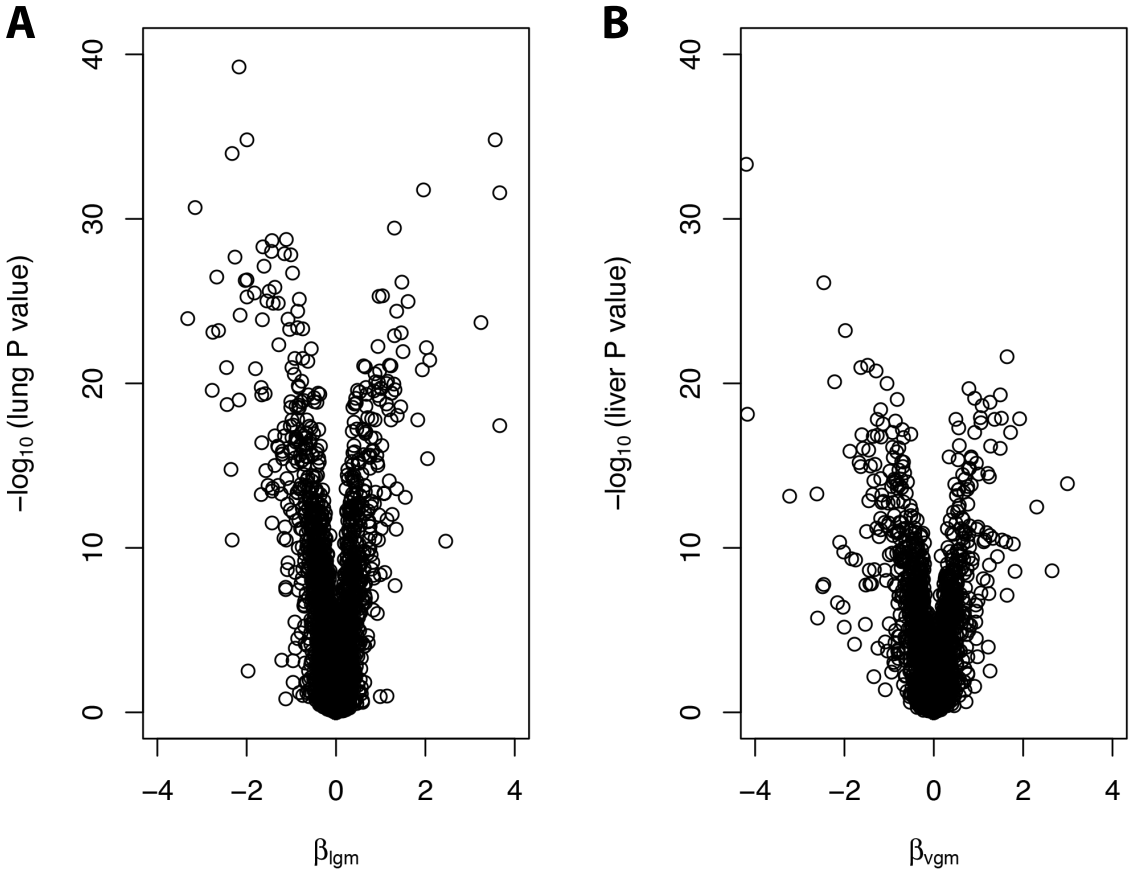

Figure IV.2: Association between β size and P value of cis-eQTLs in mouse lung and liver. The β values and P values of cis-eQTLs in mouse lung and liver tissues were derived from conventional eQTL analysis (simple linear regression). Then we selected the gene-SNP pair with minimum P value at gene level in each tissue. Volcano plots depicts the distributions of β values and $-log_{10}(P)$ of cis-eQTLs in mouse lung (A) and liver (B).

predicted by the weighted Bayesian model, the probability of posterior estimates ($\tilde{\beta}$) less than 0 was determined based on the value of $\tilde{\beta}$ and its variance in the normal distribution. We summarized the number of significant cis-eQTL at different thresholds of the probability of posterior beta ($\tilde{\beta}$) less than 0 (Table IV.2). We refer to the weighted Bayesian model we developed as tissue augmented Bayesian model of eQTL (TA-eQTL).

IV.4 Model performance assessment

To assess the performance of the developed Bayesian model, we first evaluate it on liver cis-eQTL derived from an allele specific expression (ASE) assay. Then we compared our

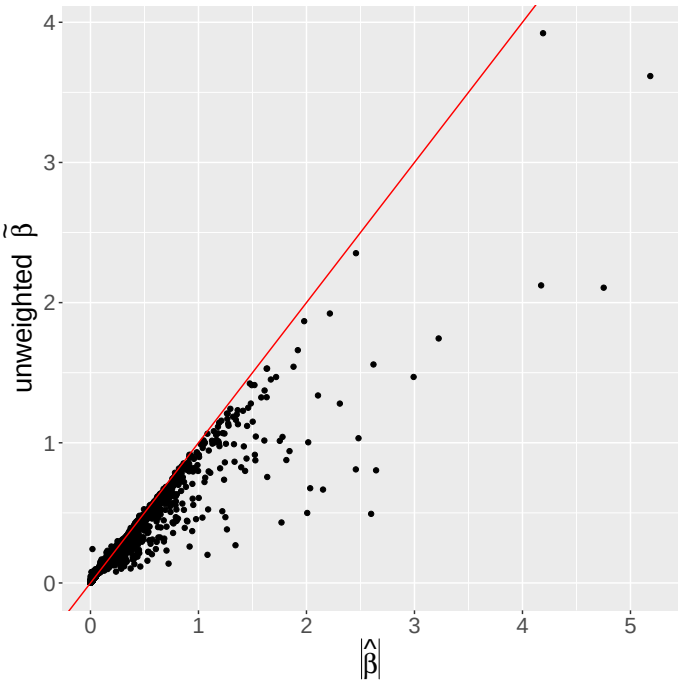

Figure IV.3: Comparison of conventional estimate of cis genotype effect on RNA expression in liver ($\left|\hat{\beta}\right|$) to the posterior estimations ($\tilde{\beta}$) in unweighted Bayesian model. Scatter plot displays the distributions of the conventional liver prediction ($\left|\hat{\beta}\right|$) and posterior estimates ($\tilde{\beta}$). The posterior estimates ($\tilde{\beta}$) were derived from the unweighted Bayesian model with prior estimation ($Z\hat{\Gamma}$) and conventional liver prediction ($\left|\hat{\beta}\right|$). The red line is a line with $slope = 1$.

Table IV.2: Summary of genes with a significant cis-eQTL based on posterior probability

Posterior probability threshold	No. of genes with significant cis-eQTL in liver (% of total)
0.05	5388 (51)
0.01	3300 (31)
0.001	2128 (20)
1e-04	1576 (15)
1e-05	1304 (12)
1e-06	1107 (10)
1e-07	970 (9)
1e-08	853 (8)
1e-09	769 (7)

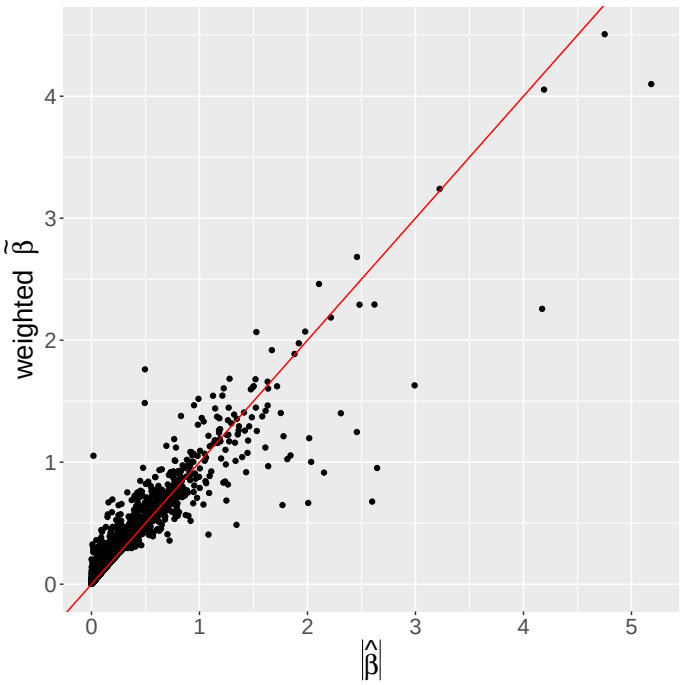

Figure IV.4: Comparison of conventional estimate of cis genotype effect on RNA expression in liver ($\left|\hat{\beta}\right|$) to the posterior estimations ($\tilde{\beta}$) in weighted Bayesian model. Scatter plot displays the distributions of the conventional liver prediction ($\left|\hat{\beta}\right|$) and posterior estimations ($\tilde{\beta}$) in weighted Bayesian model. The posterior estimations ($\tilde{\beta}$) were derived from weighted Bayesian model with prior estimation ($Z\hat{\Gamma}$) and conventional liver prediction ($\left|\hat{\beta}\right|$). The red line is a line with $slope = 1$ and $intercept = 0$.

modified Bayesian model with several existing methods in terms of sensitivity and specificity using the liver ASE set as the gold standard.

IV.4.1 Comparison of TA-eQTL model with ASE cis-eQTL

In the ASE experiment, 272 genes had significant cis-eQTL in mice with standard diet (i.e., chow-fed). The median of liver negative log P values is much larger in genes with ASE cis-eQTL than the genes without a significant cis-eQTL (Figure IV.5). The trend is maintained when comparing the lung cis-eQTL to the ASE cis-eQTL from liver, which further suggests that the association between SNP and genes are conserved between liver and lung. Of note, the median difference of negative log P values between ASE and Non-ASE groups in mouse lung is less than the one in the mouse liver.

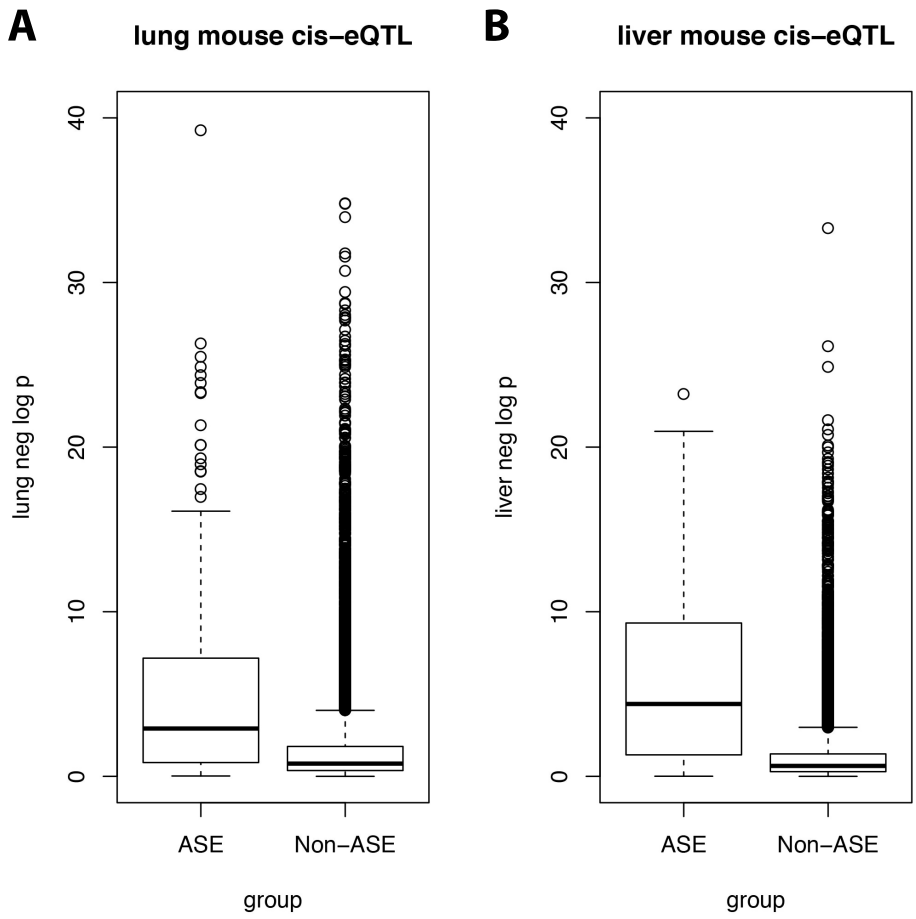

Figure IV.5: Negative log liver/lung P value distribution between ASE and Non-ASE groups. Genes were separated into two group, ASE ($n = 272$) and Non-ASE ($n = 10307$) based on the identification of allele specific expression (true-eQTL) in liver on chow-fed mice. Box plots indicate the distributions of negative log P value from conventional cis-eQTL analyses within the ASE cis-eQTL and non-ASE cis-eQTL groups in mouse lung (A) and liver (B).

IV.4.2 Comparison of TA-eQTL model with other statistical methods

Next we sorted and ranked the P values or probabilities in each method and used them as thresholds to predict "significant" or "non-significant" cis-eQTL. Then we were able to determine the sensitivity and specificity of methods based on the "ASE gold standard". Using receiver operating characteristic (ROC) curves, we compared the power and accuracy between Bayesian models and other existing approaches (Figure IV.6). Of note, the closer the ROC curve follows the top-left corner of the ROC space, the more accurate the method. The closer the ROC curve comes to the 45-degree diagonal of the ROC space, the less accurate the method. As shown in Figure IV.6, the ROC curve of lung cis-eQTL prediction is

closest to the 45-degree diagonal, which indicates that it is the least accurate among the five tested methods. Compared with the conventional liver cis-eQTL study, the three approaches incorporating lung prior knowledge (TA-eQTL method, MT approach and meta-analysis) have better performance in predicting liver cis-eQTL. The Delong's test for ROC curves further reveals that both TA-eQTL and MT methods predict cis-eQTL significantly better than the conventional liver analysis ($P_{TA-eQTL} < 0.001$, $P_{MT} = 0.008$) and conventional lung analysis ($P_{TA-eQTL} < 0.001$, $P_{MT} < 0.001$). Conventional lung analysis is just assuming that cis-eQTL in lung are also present in liver. However, the difference between the meta-analysis and the conventional liver study is not significant. In addition, we compared the TA-eQTL and MT ROC curves and their difference is not significant ($P = 0.3$).

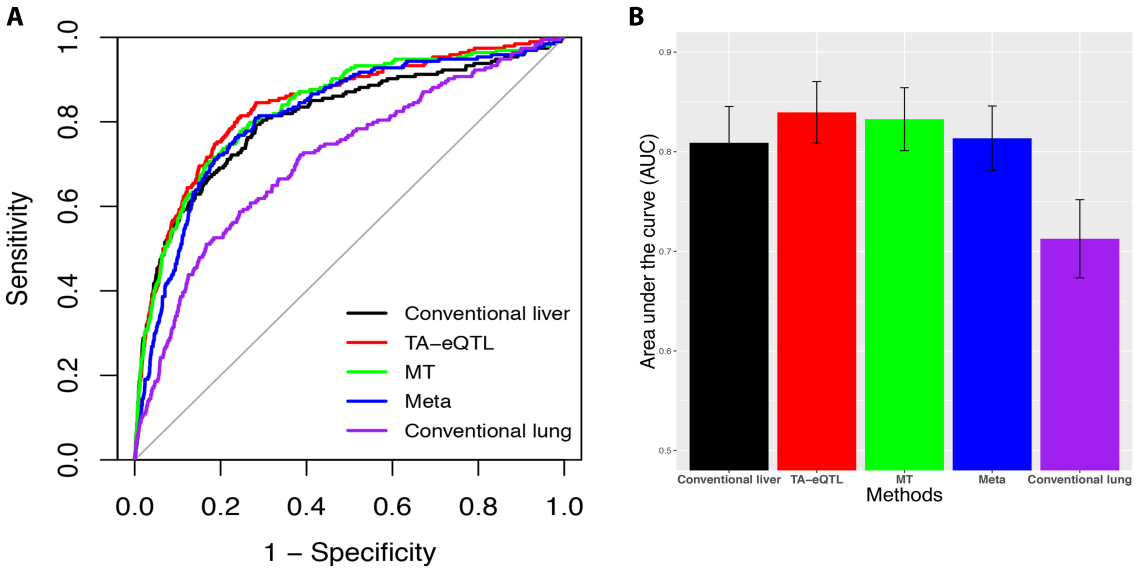

Figure IV.6: Accuracy comparison of five methods for identifying cis-eQTL in liver. (A) The black, red, green, blue and purple lines represent the ROC cures of five analysis methods: conventional liver cis-eQTL analysis (no prior), TA-eQTL method, multiple-tissue (MT) Bayesian approach, meta-analysis and lung cis-eQTL analysis. (B) The area under the ROC curves were computed following the trapezoid rule and the 95% confidence interval (CI) was determined through the bootstrap method. Each bar represents the AUC of a prediction method. The error bars represent for the 95% confidence interval.

To further quantify the performance of the five methods, we calculated the area under the curve (integral) following the trapezoid rule and determined the confidence interval based on a bootstrap strategy. As shown in supplemental Table A.3 and Figure IV.6B, the AUCs of

the three approaches incorporating lung prior knowledge (TA-eQTL method, MT approach and meta-analysis) are bigger than the AUC of the conventional liver analysis. In addition, the TA-eQTL model has a larger AUC than the MT approach and meta-analysis although pairwise comparisons among these three groups were not significant.

IV.4.3 Model performance evaluation based on sub-sampling

A major aim in developing the augmented Bayesian model is to improve the power and accuracy for cis-eQTL prediction when the sample size is small. To address the effect of sample size, we sub-sampled the liver gene dataset but maintained the prior information from the complete lung eQTL data analysis. We compared the area under ROC curves between the TA-eQTL model we developed and the other 4 approaches under different sub-samplings (10 strains, 15 strains, 20 stains and 25 strains).

According to Figure IV.7 and supplemental Table A.4, TA-eQTL is most advantageous for smaller sample sizes. For the conventional liver cis-eQTL analysis with simple linear regression, the AUC decreased quickly when the number of strains decreased. For example, if only including liver gene data from 10 BXD strains, the AUC of the basic liver model is 0.74 while it was 0.81 with the full liver dataset (30 strains). The performance of the conventional liver analysis is sensitive to the number of mouse strains. However, the AUCs of the TA-eQTL method, MT method, and meta-analysis do not decrease as much as the AUC for the conventional liver cis-eQTL prediction when sample size decreases. To better evaluate the model performances, we normalized the AUC derived from five tested methods with the one from conventional method and calculated the AUC ratio for comparison. As shown in Figure A.1, when the liver sample size decreased (less strains), the AUC ratios of the TA-eQTL method, MT approach, meta-analysis increased. These findings suggest that the three methods incorporating prior information are not as sensitive to the quantity of data as the conventional liver cis-eQTL analysis without lung information. In addition, as shown in Figure IV.7E, supplemental Figure A.1 and Table A.4, the AUC in the TA-eQTL model is significant larger than the other two methods incorporating prior lung information under all the subsetting conditions we tested (all $P < 0.001$). These results indicate that the TA-eQTL model predicts the liver cis-eQTL with higher accuracy than other tested methods,

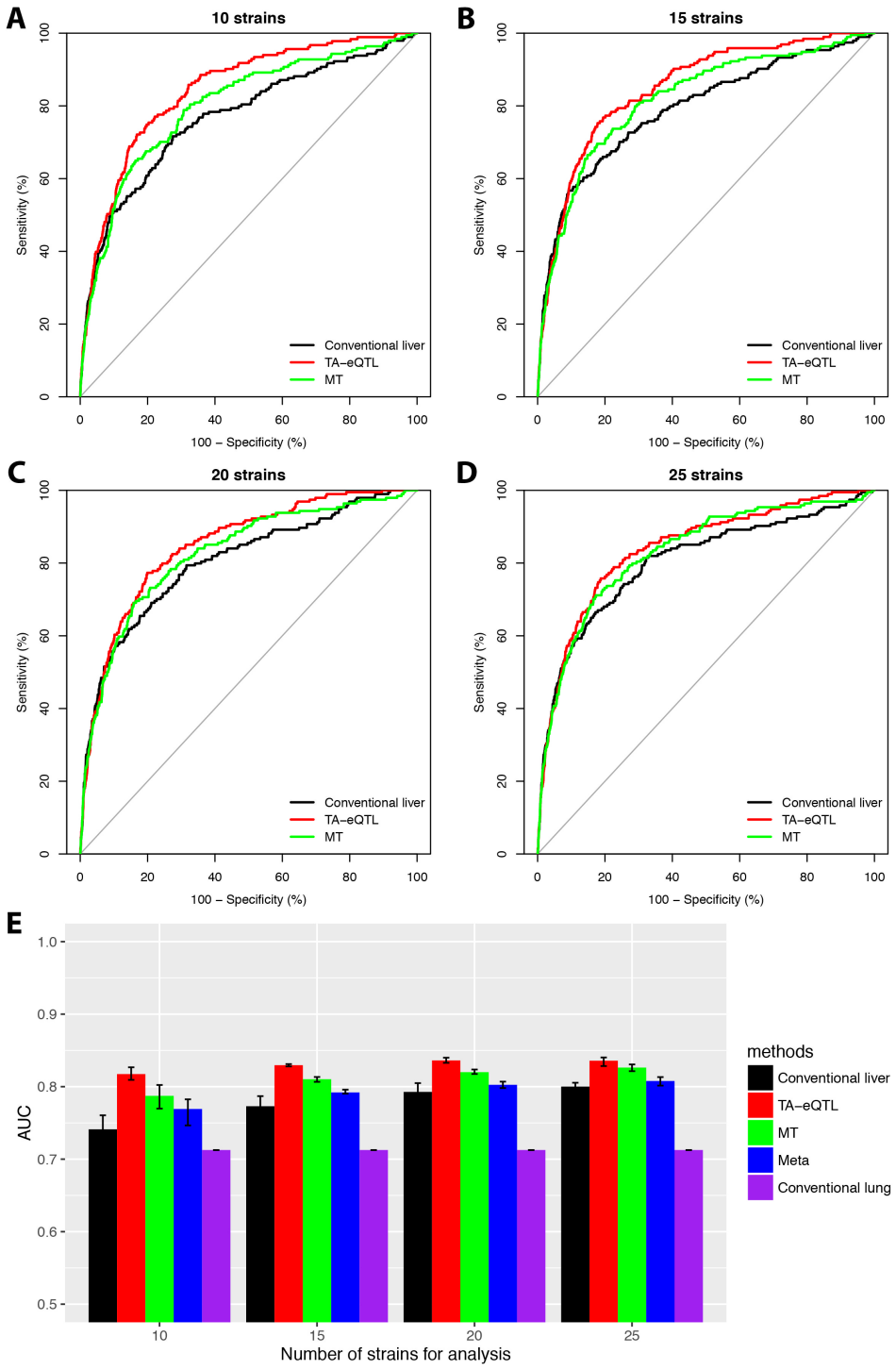

Figure IV.7: Accuracy comparison of cis-eQTL methods across different sample sizes. (A-D) The liver gene expression data were randomly subsetted to evaluate the model performances. The subsetted liver gene expression data include 10, 15, 20, and 25 strains, respectively. Each subsetting has been performed for six times with random sampling. The black, red and green lines represent the average ROC curves of three methods: conventional liver cis-eQTL analysis, TA-eQTL method, and MT approach. (E) Each colored bar chart represents the AUC of each prediction method, as indicated. The error bars represents the minimum and maximum values of AUC derived from the six times random samples.

especially when the sample data decreases. Thus, TA-eQTL method is most advantageous for smaller sample size.

CHAPTER V

DISCUSSION

V.1 Statistical discussion

In this study, we developed a tissue augmented Bayesian model of cis-eQTL (TA-eQTL) in one tissue by incorporating information from an additional tissue. Most eQTL studies have focused on the association between genetic variation and expression in a single tissue. We focus on the hypothesis that multiple tissue analyses have the potential to improve eQTL predictions (Chen and Witte, 2007; Li *et al.*, 2016; Sul *et al.*, 2013). Bayesian methods provide a natural modeling framework for eQTL analysis to take prior information into account. The prior information shared across tissues can increase the power to detect eQTLs. eQTL analyses are generally divided into two categories: gene-level analysis and SNP-level analysis (Li *et al.*, 2016). The former aims at the identification of genes with any cis-eQTL while the latter attempt to identify individual SNPs that are significantly associated with a gene. Here we focused on the identification of genes with cis-eQTL. In this study, we first assessed model performance based on liver "ASE-verified cis-eQTL" and compared the newly developed TA-eQTL model and other methods including Multiple Tissue Bayesian method (MT) and meta-analysis. We also evaluated model performance as the sample size decreased.

Our results demonstrated that both Bayesian analysis strategies (TA-eQTL and MT) significantly improved cis-eQTL gene prediction when compared with the conventional eQTL method and the meta-analysis approach, based on ROC curves and AUC. Although we did not find significant differences between the two Bayesian analysis strategies (TA-eQTL and MT) in the full dataset analysis (30 strains), we observed that the TA-eQTL method significantly improved the accuracy when sample size decreased, compared to the MT method. TA-eQTL is not as sensitive to sample size as the conventional method and other approaches. Although the ROC curves of TA-eQTL and MT methods crossed over when analyzing 25 or 30 BXD strains liver gene expression data, the TA-eQTL method has higher accuracy than the MT method with more stringent P value cutoffs.

V.2 Advantages and limitations

Compared with other existing methods, TA-eQTL method has several advantages. First, it is easy and fast to compute since TA-eQTL is in closed form and was designed to identify genes controlled by cis-eQTL and focus on the gene-SNP pair with minimum P value at that gene. Secondly, TA-eQTL could be easily applied to studies that are not perfectly matched by platforms. For example, it is not unusual to have different arrays or unequal number of probesets for multiple tissues as we do in our example. In these cases, the MT method might not work well since it can only analyze the overlapped probesets across tissues. However, our TA-eQTL method can handle these data since it pre-selects the gene-SNP pair with minimum P value at the gene level (but not the probeset level) for further Bayesian analysis. Thirdly, TA-eQTL predicts cis-eQTL gene in a more accurate way than other testing methods, especially when sample size is small.

Despite many advantages, there are some limitations of this method. One limitation is that the TA-eQTL method does not efficiently use all of the information contained in these large and complex data sets. For example, one gene could have several significant gene-SNP pairs. Another limitation of this particular study is that the ASE gene list we used as the gold standard to evaluate model performance might not be complete because it only captures genes with true cis-eQTL that also have a genetic variant within the transcribed region. Our method also identifies local eQTL that may not represent differences in expression due to one allele either being repressed or activated.

V.3 Future directions

In the future, we plan to validate our prediction with additional ASE-genes in mouse liver. We would also like to optimize the weight and try alternative prior distributions for the Bayesian model based on real data and simulation. We can also extend the tissue augmented Bayesian model to three or more tissues available in the BXD mouse panel (e.g., hippocampus, kidney and eye). In addition, we plan to develop the species augmented Bayesian model to incorporate mouse eQTL information to improve human eQTL prediction.

REFERENCES

RStudio Team (2015). *RStudio: Integrated Development Environment for R*. RStudio, Inc., Boston, MA.

Alberts R, Lu L, Williams RW, Schughart K (2011). "Genome-wide analysis of the mouse lung transcriptome reveals novel molecular gene interaction networks and cell-specific expression signatures." *Respir Res*, **12**, 61. doi:10.1186/1465-9921-12-61.

Bates D, Mächler M, Bolker B, Walker S (2015). "Fitting Linear Mixed-Effects Models Using lme4." *Journal of Statistical Software*, **67**(1), 1–48. doi:10.18637/jss.v067.i01.

Bennett BJ, Farber CR, Orozco L, Kang HM, Ghazalpour A, Siemers N, Neubauer M, Neuhaus I, Yordanova R, Guan B, Truong A, Yang Wp, He A, Kayne P, Gargalovic P, Kirchgessner T, Pan C, Castellani LW, Kostem E, Furlotte N, Drake TA, Eskin E, Lusis AJ (2010). "A high-resolution association mapping panel for the dissection of complex traits in mice." *Genome Res*, **20**(2), 281–90. doi:10.1101/gr.099234.109.

Blauwendraat C, Francescatto M, Gibbs JR, Jansen IE, Simón-Sánchez J, Hernandez DG, Dillman AA, Singleton AB, Cookson MR, Rizzu P, Heutink P (2016). "Comprehensive promoter level expression quantitative trait loci analysis of the human frontal lobe." *Genome Med*, **8**(1), 65. doi:10.1186/s13073-016-0320-1.

Carneiro AMD, Airey DC, Thompson B, Zhu CB, Lu L, Chesler EJ, Erikson KM, Blakely RD (2009). "Functional coding variation in recombinant inbred mouse lines reveals multiple serotonin transporter-associated phenotypes." *Proc Natl Acad Sci U S A*, **106**(6), 2047–52. doi:10.1073/pnas.0809449106.

Chen GK, Witte JS (2007). "Enriching the analysis of genomewide association studies with hierarchical modeling." *Am J Hum Genet*, **81**(2), 397–404. doi:10.1086/519794.

Chesler EJ, Lu L, Wang J, Williams RW, Manly KF (2004). "WebQTL: rapid exploratory analysis of gene expression and genetic networks for brain and behavior." *Nat Neurosci*, **7**(5), 485–6. doi:10.1038/nn0504-485.

Cookson W, Liang L, Abecasis G, Moffatt M, Lathrop M (2009). "Mapping complex disease traits with global gene expression." *Nat Rev Genet*, **10**(3), 184–94. doi:10.1038/nrg2537.

Cubillos FA, Coustham V, Loudet O (2012). "Lessons from eQTL mapping studies: non-coding regions and their role behind natural phenotypic variation in plants." *Curr Opin Plant Biol*, **15**(2), 192–8. doi:10.1016/j.pbi.2012.01.005.

Dahl DB (2016). *xtable: Export Tables to LaTeX or HTML*. R package version 1.8-2.

Das A, Morley M, Moravec CS, Tang WHW, Hakonarson H, MAGNet Consortium, Margulies KB, Cappola TP, Jensen S, Hannenhalli S (2015). "Bayesian integration of genetics and epigenetics detects causal regulatory SNPs underlying expression variability." *Nat Commun*, **6**, 8555. doi:10.1038/ncomms9555.

Davis RC, van Nas A, Castellani LW, Zhao Y, Zhou Z, Wen P, Yu S, Qi H, Rosales M, Schadt EE, Broman KW, Péterfy M, Lusis AJ (2012). "Systems genetics of susceptibility to obesity-induced diabetes in mice." *Physiol Genomics*, **44**(1), 1–13. doi: 10.1152/physiolgenomics.00003.2011.

DeLong ER, DeLong DM, Clarke-Pearson DL (1988). "Comparing the areas under two or more correlated receiver operating characteristic curves: a nonparametric approach." *Biometrics*, **44**(3), 837–45.

Dixon AL, Liang L, Moffatt MF, Chen W, Heath S, Wong KCC, Taylor J, Burnett E, Gut I, Farrall M, Lathrop GM, Abecasis GR, Cookson WOC (2007). "A genome-wide association study of global gene expression." *Nat Genet*, **39**(10), 1202–7. doi:10.1038/ng2109.

Dowle M, Srinivasan A, Short T, with contributions from R Saporta SL, Antonyan E (2015). *data.table: Extension of Data.frame*. R package version 1.9.6.

Durinck S, Moreau Y, Kasprzyk A, Davis S, De Moor B, Brazma A, Huber W (2005). "BioMart and Bioconductor: a powerful link between biological databases and microarray data analysis." *Bioinformatics*, **21**(16), 3439–40. doi:10.1093/bioinformatics/bti525.

Flutre T, Wen X, Pritchard J, Stephens M (2013). "A statistical framework for joint eQTL analysis in multiple tissues." *PLoS Genet*, **9**(5), e1003486. doi:10.1371/journal.pgen.1003486.

Fraser HB, Moses AM, Schadt EE (2010). "Evidence for widespread adaptive evolution of gene expression in budding yeast." *Proc Natl Acad Sci U S A*, **107**(7), 2977–82. doi: 10.1073/pnas.0912245107.

Gelman A, Carlin JB, Stern HS, Rubin DB (2014). *Bayesian data analysis*, volume 2. Chapman & Hall/CRC Boca Raton, FL, USA.

Gerrits A, Li Y, Tesson BM, Bystrykh LV, Weersing E, Ausema A, Dontje B, Wang X, Breitling R, Jansen RC, de Haan G (2009). "Expression quantitative trait loci are highly sensitive to cellular differentiation state." *PLoS Genet*, **5**(10), e1000692. doi:10.1371/journal.pgen.1000692.

Göring HHH, Curran JE, Johnson MP, Dyer TD, Charlesworth J, Cole SA, Jowett JBM, Abraham LJ, Rainwater DL, Comuzzie AG, Mahaney MC, Almasy L, MacCluer JW, Kissebah AH, Collier GR, Moses EK, Blangero J (2007). "Discovery of expression QTLs using large-scale transcriptional profiling in human lymphocytes." *Nat Genet*, **39**(10), 1208–16. doi:10.1038/ng2119.

Hindorff LA, Sethupathy P, Junkins HA, Ramos EM, Mehta JP, Collins FS, Manolio TA (2009). "Potential etiologic and functional implications of genome-wide association loci for human diseases and traits." *Proc Natl Acad Sci U S A*, **106**(23), 9362–7. doi:10.1073/pnas.0903103106.

Hrdlickova B, de Almeida RC, Borek Z, Withoff S (2014). "Genetic variation in the non-coding genome: Involvement of micro-RNAs and long non-coding RNAs in disease." *Biochim Biophys Acta*, **1842**(10), 1910–1922. doi:10.1016/j.bbadis.2014.03.011.

Imholte GC, Scott-Boyer MP, Labbe A, Deschepper CF, Gottardo R (2013). "iBMQ: a R/Bioconductor package for integrated Bayesian modeling of eQTL data." *Bioinformatics*, **29**(21), 2797–8. doi:10.1093/bioinformatics/btt485.

Jansen RC, Nap JP (2001). "Genetical genomics: the added value from segregation." *Trends Genet*, **17**(7), 388–91.

Jurasinski G, Koebsch F, Guenther A, Beetz S (2014). *flux: Flux rate calculation from dynamic closed chamber measurements*. R package version 0.3-0.

Keane TM, Goodstadt L, Danecek P, White MA, Wong K, Yalcin B, Heger A, Agam A, Slater G, Goodson M, Furlotte NA, Eskin E, Nellåker C, Whitley H, Cleak J, Janowitz D, Hernandez-Pliego P, Edwards A, Belgard TG, Oliver PL, McIntyre RE, Bhomra A, Nicod J, Gan X, Yuan W, van der Weyden L, Steward CA, Bala S, Stalker J, Mott R, Durbin R, Jackson IJ, Czechanski A, Guerra-Assunção JA, Donahue LR, Reinholdt LG, Payseur BA, Ponting CP, Birney E, Flint J, Adams DJ (2011). "Mouse genomic variation and its effect on phenotypes and gene regulation." *Nature*, **477**(7364), 289–94. doi:10.1038/nature10413.

Kulis B (2012). "Bayesain Linear Regression." *CSE 788.94: Topics in Machine Learning*.

Lagarrigue S, Martin L, Hormozdiari F, Roux PF, Pan C, van Nas A, Demeure O, Cantor R, Ghazalpour A, Eskin E, Lusis AJ (2013). "Analysis of allele-specific expression in mouse liver by RNA-Seq: a comparison with Cis-eQTL identified using genetic linkage." *Genetics*, **195**(3), 1157–66. doi:10.1534/genetics.113.153882.

Laoutidis ZG, Luckhaus C (2015). "The Liptak-Stouffer test for meta-analyses." *Biol Psychiatry*, **77**(1), e1–2. doi:10.1016/j.biopsych.2013.11.033.

Lenth RV (2016). "Least-Squares Means: The R Package lsmeans." *Journal of Statistical Software*, **69**(1), 1–33. doi:10.18637/jss.v069.i01.

Li G, Shabalin AA, Rusyn I (2016). "An Empirical Bayes Approach for Multiple Tissue eQTL Analysis." *arXiv:1311.2948 [stat.ME]*.

Nica AC, Dermitzakis ET (2008). "Using gene expression to investigate the genetic basis of complex disorders." *Hum Mol Genet*, **17**(R2), R129–34. doi:10.1093/hmg/ddn285.

Nica AC, Dermitzakis ET (2013). "Expression quantitative trait loci: present and future." *Philos Trans R Soc Lond B Biol Sci*, **368**(1620), 20120362. doi:10.1098/rstb.2012.0362.

Pandey AK, Williams RW (2014). "Genetics of gene expression in CNS." *Int Rev Neurobiol*, **116**, 195–231. doi:10.1016/B978-0-12-801105-8.00008-4.

Phillips TJ, Huson M, Gwiazdon C, Burkhart-Kasch S, Shen EH (1995). "Effects of acute and repeated ethanol exposures on the locomotor activity of BXD recombinant inbred mice." *Alcohol Clin Exp Res*, **19**(2), 269–78.

R Core Team (2015). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria.

Ricaño-Ponce I, Wijmenga C (2013). "Mapping of immune-mediated disease genes." *Annu Rev Genomics Hum Genet*, **14**, 325–53. doi:10.1146/annurev-genom-091212-153450.

Robin X, Turck N, Hainard A, Tiberti N, Lisacek F, Sanchez JC, Müller M (2011). "pROC: an open-source package for R and S+ to analyze and compare ROC curves." *BMC Bioinformatics*, **12**, 77.

Rockman MV, Kruglyak L (2006). "Genetics of global gene expression." *Nat Rev Genet*, **7**(11), 862–72. doi:10.1038/nrg1964.

Ronald J, Brem RB, Whittle J, Kruglyak L (2005). "Local regulatory variation in Saccharomyces cerevisiae." *PLoS Genet*, **1**(2), e25. doi:10.1371/journal.pgen.0010025.

Schadt EE, Molony C, Chudin E, Hao K, Yang X, Lum PY, Kasarskis A, Zhang B, Wang S, Suver C, Zhu J, Millstein J, Sieberts S, Lamb J, GuhaThakurta D, Derry J, Storey JD, Avila-Campillo I, Kruger MJ, Johnson JM, Rohl CA, van Nas A, Mehrabian M, Drake TA, Lusis AJ, Smith RC, Guengerich FP, Strom SC, Schuetz E, Rushmore TH, Ulrich R (2008). "Mapping the genetic architecture of gene expression in human liver." *PLoS Biol*, **6**(5), e107. doi:10.1371/journal.pbio.0060107.

Scott-Boyer MP, Imholte GC, Tayeb A, Labbe A, Deschepper CF, Gottardo R (2012). "An integrated hierarchical Bayesian model for multivariate eQTL mapping." *Stat Appl Genet Mol Biol*, **11**(4). doi:10.1515/1544-6115.1760.

Serre D, Gurd S, Ge B, Sladek R, Sinnett D, Harmsen E, Bibikova M, Chudin E, Barker DL, Dickinson T, Fan JB, Hudson TJ (2008). "Differential allelic expression in the human genome: a robust approach to identify genetic and epigenetic cis-acting mechanisms regulating gene expression." *PLoS Genet*, **4**(2), e1000006. doi:10.1371/journal.pgen.1000006.

Shabalin AA (2012). "Matrix eQTL: ultra fast eQTL analysis via large matrix operations." *Bioinformatics*, **28**(10), 1353–8. doi:10.1093/bioinformatics/bts163.

Skelly DA, Johansson M, Madeoy J, Wakefield J, Akey JM (2011). "A powerful and flexible statistical framework for testing hypotheses of allele-specific gene expression from RNA-seq data." *Genome Res*, **21**(10), 1728–37. doi:10.1101/gr.119784.110.

Stegle O, Parts L, Durbin R, Winn J (2010). "A Bayesian framework to account for complex non-genetic factors in gene expression levels greatly increases power in eQTL studies." *PLoS Comput Biol*, **6**(5), e1000770. doi:10.1371/journal.pcbi.1000770.

Stephens M, Balding DJ (2009). "Bayesian statistical methods for genetic association studies." *Nat Rev Genet*, **10**(10), 681–90. doi:10.1038/nrg2615.

Stouffer S DeVinney L SE (1949). "The American soldier: Adjustment during army life." *Princeton University Press*, **Vol. 1.**

Sul JH, Han B, Ye C, Choi T, Eskin E (2013). "Effectively identifying eQTLs from multiple tissues by combining mixed model and meta-analytic approaches." *PLoS Genet*, **9**(6), e1003491. doi:10.1371/journal.pgen.1003491.

T L (1958). "On the combination of independent tests." *Magyar Tud Akad Mat Kutato Int Közl*, (171-196).

Tabakoff B, Saba L, Kechris K, Hu W, Bhave SV, Finn DA, Grahame NJ, Hoffman PL (2008). "The genomic determinants of alcohol preference in mice." *Mamm Genome*, **19**(5), 352–65. doi:10.1007/s00335-008-9115-z.

Team RC, Wuertz D, Setz T, Chalabi Y (2014). *fBasics: Rmetrics - Markets and Basic Statistics*. R package version 3011.87.

Veyrieras JB, Kudaravalli S, Kim SY, Dermitzakis ET, Gilad Y, Stephens M, Pritchard JK (2008). "High-resolution mapping of expression-QTLs yields insight into human gene regulation." *PLoS Genet*, **4**(10), e1000214. doi:10.1371/journal.pgen.1000214.

Wang J, Williams RW, Manly KF (2003). "WebQTL: web-based complex trait analysis." *Neuroinformatics*, **1**(4), 299–308. doi:10.1385/NI:1:4:299.

Webster JA, Gibbs JR, Clarke J, Ray M, Zhang W, Holmans P, Rohrer K, Zhao A, Marlowe L, Kaleem M, McCorquodale 3rd DS, Cuello C, Leung D, Bryden L, Nath P, Zismann VL, Joshipura K, Huentelman MJ, Hu-Lince D, Coon KD, Craig DW, Pearson JV, NACC-Neuropathology Group, Heward CB, Reiman EM, Stephan D, Hardy J, Myers AJ (2009). "Genetic control of human brain transcript expression in Alzheimer disease." *Am J Hum Genet*, **84**(4), 445–58. doi:10.1016/j.ajhg.2009.03.011.

Whitlock MC (2005). "Combining probability from independent tests: the weighted Z-method is superior to Fisher's approach." *J Evol Biol*, **18**(5), 1368–73. doi:10.1111/j.1420-9101.2005.00917.x.

Wickham H (2009). *ggplot2: Elegant Graphics for Data Analysis*. Springer-Verlag New York. ISBN 978-0-387-98140-6.

Wickham H (2011). "The Split-Apply-Combine Strategy for Data Analysis." *Journal of Statistical Software*, **40**(1), 1–29.

Wittkopp PJ, Haerum BK, Clark AG (2004). "Evolutionary changes in cis and trans gene regulation." *Nature*, **430**(6995), 85–8. doi:10.1038/nature02698.

Zhang X, Huang S, Sun W, Wang W (2012). "Rapid and robust resampling-based multiple-testing correction with application in a genome-wide expression quantitative trait loci study." *Genetics*, **190**(4), 1511–20. doi:10.1534/genetics.111.137737.

APPENDIX A

Supplemental results

Table A.1: Summary of overlap of lung and liver cis eQTL: observed vs expected.

	P value threshold	P value of Chisq test	observed shared	expected shared	Ratio
1	0.95	0.8250	10296	10292	1.00
2	0.90	0.7219	9967	9958	1.00
3	0.85	0.5941	9624	9609	1.00
4	0.80	0.2252	9257	9218	1.00
5	0.75	0.1270	8817	8761	1.01
6	0.70	0.0128	8381	8282	1.01
7	0.65	0.0010	7924	7783	1.02
8	0.60	<0.0001	7458	7272	1.03
9	0.55	<0.0001	6949	6728	1.03
10	0.50	<0.0001	6435	6181	1.04
11	0.45	<0.0001	5914	5621	1.05
12	0.40	<0.0001	5366	5018	1.07
13	0.35	<0.0001	4825	4445	1.09
14	0.30	<0.0001	4297	3866	1.11
15	0.25	<0.0001	3800	3283	1.16
16	0.20	<0.0001	3273	2701	1.21
17	0.15	<0.0001	2738	2121	1.29
18	0.10	<0.0001	2235	1561	1.43
19	0.05	<0.0001	1673	975	1.72
20	0.01	<0.0001	1028	428	2.40
21	0.001	<0.0001	702	208	3.38
22	1e-04	<0.0001	515	120	4.29
23	1e-05	<0.0001	389	75	5.17
24	1e-06	<0.0001	280	47	5.96
25	1e-07	<0.0001	216	31	6.97
26	5e-08	<0.0001	190	27	7.15

$Ratio = \frac{observed\ shared}{expected\ shared}$

Table A.2: Summary of β predictions in the unweighted Bayesian model

	$\tilde{\beta}$	$\hat{\beta}$	$z\hat{\gamma}$
Mean	0.10	0.11	0.11
Stdev	0.16	0.21	0.09
Median	0.05	0.05	0.08
Minimum	0.00	0.00	0.06
Maximum	3.92	5.18	1.17

Table A.3: AUC comparison among five predicting methods

	AUC	CI
Conventional liver	0.81	(0.77, 0.85)
Bayesian	0.84	(0.81, 0.87)
MT	0.83	(0.80, 0.86)
Meta	0.81	(0.78, 0.85)
Conventional lung	0.71	(0.67, 0.75)

Table A.4: AUC comparison with subsetted dataset

	Subsample (10 strains)		Subsample (15 strains)		Subsample (20 strains)		Subsample (25 strains)		Full sample	
	AUC	Ratio	AUC	Ratio	AUC	Ratio	AUC	Ratio	AUC	Ratio
Conventional liver	0.74	1.00	0.77	1.00	0.79	1.00	0.80	1.00	0.81	1.00
TA-eQTL	0.82	1.10	0.83	1.07	0.84	1.05	0.84	1.04	0.84	1.04
MT	0.79	1.06	0.81	1.05	0.82	1.03	0.83	1.03	0.83	1.03
Meta	0.77	1.04	0.79	1.02	0.80	1.01	0.81	1.01	0.81	1.01
Conventional lung	0.71	0.96	0.71	0.92	0.71	0.90	0.71	0.89	0.71	0.88

$Ratio = \frac{AUC}{AUC_{\text{Conventional liver}}}$

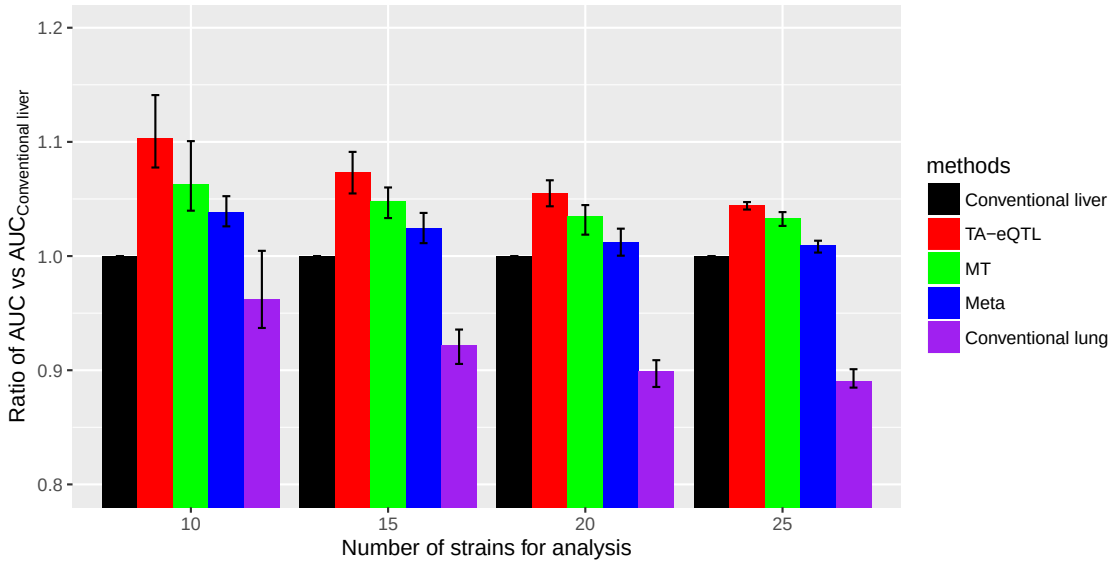

Figure A.1: AUC ratios of cis-eQTL methods across different sample sizes. To evaluate the model performances, the liver gene expression data were randomly subsetted. The subsetted liver gene expression data include 10 strains, 15 strains, 20 strains, and 25 strains, respectively. The lung cis-eQTL analysis are performed on 45 strains all the time. The area under the ROC curves were computed following the trapezoid rule. The AUC ratios to the AUC from the conventional method in liver are reported. As indicated, each colored bar chart represents the AUC of each prediction methods: conventional liver cis-eQTL analysis (Conventional liver), TA-eQTL method, MT method, Meta-analysis and lung cis-eQTL analysis (Conventional lung). The error bars represents the minimum and maximum values of AUC derived from six random samples at each subsetting.

APPENDIX B

R codes

Unless otherwise noted, all codes are written in the R statistical programming language (R Core Team, 2015).

B.1 Step 0 - Data Pre-processing

```
# Prepare liver gene location and snp location for eQTL analysis
rm(list = ls())
setwd("/Volumes/Transcend/Thesis_project/eQTL data/1.1.liver.gene.snp")
BXD.geno <- read.table(file = "BXD-3.geno.txt", header = T)
# recode SNP, 'B to 0, D to 1'
BXD.geno1 <- as.data.frame(sapply(BXD.geno, gsub, pattern = "B",
    replacement = "0"))
BXD.geno2 <- as.data.frame(sapply(BXD.geno1, gsub, pattern = "H",
    replacement = "NA"))
BXD.geno3 <- as.data.frame(sapply(BXD.geno2, gsub, pattern = "D",
    replacement = "1"))
BXD.geno4 <- as.data.frame(sapply(BXD.geno3, gsub, pattern = "U",
    replacement = "NA"))
# SNPloc.txt was downloaded from BioMart-Ensembl website website
SNPloc <- read.table(file = "SNPloc1.txt", header = T)
SNPloc <- SNPloc[!duplicated(SNPloc$SNP), ]
SNPlibrary <- unique(SNPloc$SNP)
BXD.geno5 <- BXD.geno4[BXD.geno4$Locus %in% SNPlibrary, ]
BXD.geno.SNP <- BXD.geno5[, c(2, 5:97)]
BXD.geno.SNP <- BXD.geno.SNP[order(BXD.geno.SNP$Locus), ]
BXD.geno.loc <- SNPloc[SNPloc$SNP %in% BXD.geno.SNP$Locus, ]
BXD.geno.loc <- BXD.geno.loc[order(BXD.geno.loc$SNP), ]
# Reformat mouse liver gene expression for Matrix eqtl analysis
mouse.liver.expression <- read.table("GN373_GSE16780_UCLA_Hybrid_MDP_
    Liver_Affy_HT_M430A_Sep11_RMA_Z-Score_Average.txt",
     comment.char = "#", header = TRUE, sep = "\t", )
mouse.liver.expression <- as.data.frame(sapply(mouse.liver.expression,
```

```
    gsub, pattern = "_at_A", replacement = "_at"))
# Creat the strain library with known SNP
BXD.geno.SNP.library <- colnames(BXD.geno.SNP)
mouse.liver.expression.eqtl <- mouse.liver.expression[, which(colnames(
    mouse.liver.expression) %in% BXD.geno.SNP.library)]
# reorder stain column names
mouse.liver.expression.eqtl <- mouse.liver.expression.eqtl[, order(
    colnames(mouse.liver.expression.eqtl))]
mouse.liver.expression.eqtl$ProbeSet <- mouse.liver.expression$ProbeSet
mouse.liver.expression.eqtl <- mouse.liver.expression.eqtl[, c(31, 1:30)
    ]
# creat strain library with liver expression data
BXD.geno.strain.library <- colnames(mouse.liver.expression.eqtl)
# select SNP on the strains which has gene expression data available
BXD.geno.SNP1 <- BXD.geno.SNP[, which(colnames(BXD.geno.SNP) %in% BXD.
    geno.strain.library)]
BXD.geno.SNP1 <- BXD.geno.SNP1[, order(colnames(BXD.geno.SNP1))]
BXD.geno.SNP1$Locus <- BXD.geno.SNP$Locus
BXD.geno.SNP.eqtl <- BXD.geno.SNP1[, c(31, 1:30)]
BXD.geno.SNP.eqtl <- BXD.geno.SNP.eqtl[order(BXD.geno.SNP.eqtl$Locus), ]
# write BXD SNP genotypes for eqtl analysis
write.table(BXD.geno.SNP.eqtl, file = "2016-09-08 BXD.geno.SNP.eqtl.for.
    liver.txt", sep = "\t", row.names = FALSE, quote = FALSE)
# check dimensions to make sure they match
dim(BXD.geno.loc)
# write BXD SNP location for eqtl analysis
write.table(BXD.geno.loc, file = "2016-09-08 BXD.geno.loc.eqtl.for.liver
    .txt", sep = "\t", row.names = FALSE, quote = FALSE)
# Affy_moe430a.txt was downloaded from BioMart-Ensembl website
mouse430a <- read.table(file = "Affy_moe430a1.txt", header = T)
mouse430a <- mouse430a[!duplicated(mouse430a$probeset), ]
liver.probeset.position.library <- mouse430a$probeset
# subset mouse liver expression data with known gene location
```

```
mouse.liver.expression.eqtl.position <- mouse.liver.expression.eqtl[
    mouse.liver.expression.eqtl$ProbeSet %in%
     liver.probeset.position.library, ]
mouse.liver.expression.eqtl.position <- mouse.liver.expression.eqtl.
    position[order(mouse.liver.expression.eqtl.position$ProbeSet), ]
# write mouse liver gene expression data for eqtl analsis
write.table(mouse.liver.expression.eqtl.position, file = "2016-09-08
    mouse.liver.expression.eqtl.txt",sep = "\t", row.names = FALSE,
    quote = FALSE)
liver.gene.loc <- mouse430a[mouse430a$probeset %in% mouse.liver.
    expression.eqtl.position$ProbeSet, ]
liver.gene.loc <- liver.gene.loc[order(liver.gene.loc$probeset), ]
write.table(liver.gene.loc, file = "2016-09-08 liver.gene.loc.txt", sep
    = "\t", row.names = FALSE, quote = FALSE)
```

```r
# Prepare lung gene location and snp location for eQTL analysis
rm(list = ls())
setwd("/Volumes/Transcend/Thesis_project/eQTL data/1.2.lung.gene.snp")
BXD.geno <- read.table(file = "BXD-3.geno.txt", header = T)
# recode SNP, 'B to 0, D to 1'
BXD.geno1 <- as.data.frame(sapply(BXD.geno, gsub, pattern = "B",
    replacement = "0"))
BXD.geno2 <- as.data.frame(sapply(BXD.geno1, gsub, pattern = "H",
    replacement = "NA"))
BXD.geno3 <- as.data.frame(sapply(BXD.geno2, gsub, pattern = "D",
    replacement = "1"))
BXD.geno4 <- as.data.frame(sapply(BXD.geno3, gsub, pattern = "U",
    replacement = "NA"))
# SNPloc.txt was downloaded from BioMart-Ensembl website website
SNPloc <- read.table(file = "SNPloc1.txt", header = T)
SNPloc <- SNPloc[!duplicated(SNPloc$SNP), ]
SNPlibrary <- unique(SNPloc$SNP)
BXD.geno5 <- BXD.geno4[BXD.geno4$Locus %in% SNPlibrary, ]
BXD.geno.loc <- SNPloc[SNPloc$SNP %in% BXD.geno5$Locus, ]
BXD.geno.loc <- BXD.geno.loc[order(BXD.geno.loc$SNP), ]
# load lung expression data
mouse.lung.expression <- read.csv("GN160_DataAvgAnnot.rev0614.csv",
    na.strings = c("", "NA"), header = TRUE, )
BXD.geno.SNP.library <- colnames(BXD.geno5)
mouse.lung.expression.eqtl <- mouse.lung.expression[, which(colnames(
    mouse.lung.expression) %in% BXD.geno.SNP.library)]
mouse.lung.expression.eqtl$Chr <- NULL
mouse.lung.expression.eqtl$Mb <- NULL
# reorder stain column names
mouse.lung.expression.eqtl <- mouse.lung.expression.eqtl[, order(
    colnames(mouse.lung.expression.eqtl))]
mouse.lung.expression.eqtl$ProbeSet <- mouse.lung.expression$ProbeSet
mouse.lung.expression.eqtl <- mouse.lung.expression.eqtl[, c(46, 1:45)]
```

```
# select SNP on the strains which has gene expression data available
BXD.geno6 <- BXD.geno5[, which(colnames(BXD.geno5) %in% colnames(mouse.
    lung.expression.eqtl))]
BXD.geno6 <- BXD.geno6[, order(colnames(BXD.geno6))]
BXD.geno6$Locus <- BXD.geno5$Locus
BXD.geno.SNP.eqtl <- BXD.geno6[, c(46, 1:45)]
BXD.geno.SNP.eqtl <- BXD.geno.SNP.eqtl[order(BXD.geno.SNP.eqtl$Locus), ]
# write BXD SNP genotypes for eqtl analysis
write.table(BXD.geno.SNP.eqtl, file = "2016-09-08 BXD.geno.SNP.eqtl.for.
    lung.txt", sep = "\t", row.names = FALSE, quote = FALSE)
# write BXD SNP location for eqtl analysis
write.table(BXD.geno.loc, file = "2016-09-08 BXD.geno.loc.eqtl.for.lung.
    txt", sep = "\t", row.names = FALSE, quote = FALSE)
mouse430 <- read.table(file = "Affy mouse4302.txt", header = T)
mouse430 <- mouse430[!duplicated(mouse430$probeset), ]
lung.probeset.position.library <- mouse430$probeset
# subset mouse lung expression data with known gene location
mouse.lung.expression.eqtl.position <- mouse.lung.expression.eqtl[mouse.
    lung.expression.eqtl$ProbeSet %in% lung.probeset.position.library, ]
mouse.lung.expression.eqtl.position <- mouse.lung.expression.eqtl.
    position[order(mouse.lung.expression.eqtl.position$ProbeSet), ]
mouse.lung.expression.eqtl.position <- mouse.lung.expression.eqtl.
    position[order(mouse.lung.expression.eqtl.position$ProbeSet), ]
# write mouse lung gene expression data for eqtl analsis
write.table(mouse.lung.expression.eqtl.position, file = "2016-09-08
    mouse.lung.expression.eqtl.txt", sep = "\t", row.names = FALSE,
    quote = FALSE)
lung.gene.loc <- mouse430[mouse430$probeset %in% mouse.lung.expression.
    eqtl.position$ProbeSet, ]
lung.gene.loc <- lung.gene.loc[order(lung.gene.loc$probeset), ]
write.table(lung.gene.loc, file = "2016-09-08 lung.gene.loc.txt", sep =
    "\t", row.names = FALSE, quote = FALSE)
```

B.2 Step 1 - Calculate eQTL

```
rm(list = ls())
gc()
# set directory
setwd("/Volumes/Transcend/Thesis_project/eQTL data")
### code good for subsetting dataset analysis however,
### if defined sebsetn=30, analyze all the data
sebsetn <- 30  # full liver dataset has 30 strains
# subset liver gene expression dataset
mouse.liver.expression.eqtl <- read.table(file = "2016-09-08 mouse.liver
    .expression.eqtl.txt", header = T)
set.seed(50)
sub.mouse.liver.expression.eqtl <- mouse.liver.expression.eqtl[, c(1,
    sample(2:dim(mouse.liver.expression.eqtl)[2], sebsetn, replace =
        FALSE))]
write.table(sub.mouse.liver.expression.eqtl, file = "sub.mouse.liver.
    expression.eqtl.txt", sep = "\t", row.names = FALSE, quote = FALSE)
# subset liver snp expression data
BXD.geno.SNP.eqtl.for.liver <- read.table(file = "2016-09-08 BXD.geno.
    SNP.eqtl.for.liver.txt", header = T)
set.seed(50)
sub.BXD.geno.SNP.eqtl.for.liver <- BXD.geno.SNP.eqtl.for.liver[, c(1,
    sample(2:dim(BXD.geno.SNP.eqtl.for.liver)[2], sebsetn, replace =
        FALSE))]
write.table(sub.BXD.geno.SNP.eqtl.for.liver, file = "sub.BXD.geno.SNP.
    eqtl.for.liver.txt", sep = "\t", row.names = FALSE, quote = FALSE)
### liver eqtl analysis
library("MatrixEQTL")
library(xtable)
base.dir <- "/Volumes/Transcend/Thesis_project/Subsetted_liver"
# Linear model to use, modelANOVA, modelLINEAR, or modelLINEAR_CROSS
useModel <- modelLINEAR
# Genotype file name
```

```
SNP_file_name <- paste(base.dir, "/sub.BXD.geno.SNP.eqtl.for.liver.txt",
     sep = "")
snps_location_file_name <- paste(base.dir, "/2016-09-08 BXD.geno.loc.
    eqtl.for.liver.txt", sep = "")
# Gene expression file name
expression_file_name <- paste(base.dir, "/sub.mouse.liver.expression.
    eqtl.txt", sep = "")
gene_location_file_name <- paste(base.dir, "/2016-09-08 liver.gene.loc.
    txt", sep = "")
# Covariates file name Set to character() for no covariates
covariates_file_name <- character()
# Output file name
output_file_name_cis <- tempfile()
output_file_name_tra <- tempfile()
# Only associations significant at this level will be saved
pvOutputThreshold_cis <- 1
pvOutputThreshold_tra <- 5e-15
# Error covariance matrix Set to numeric() for identity.
errorCovariance <- numeric()
# errorCovariance = read.table('Sample_Data/errorCovariance.txt');
# Distance for local gene-SNP pairs
cisDist <- 1e+06
## Load genotype data
snps <- SlicedData$new()
snps$fileDelimiter <- "\t"
snps$fileOmitCharacters <- "NA"
snps$fileSkipRows <- 1
snps$fileSkipColumns <- 1
snps$fileSliceSize <- 2000
snps$LoadFile(SNP_file_name)
## Load gene expression data
gene <- SlicedData$new()
gene$fileDelimiter <- "\t"
```

```
gene$fileOmitCharacters <- "NA"
gene$fileSkipRows <- 1
gene$fileSkipColumns <- 1
gene$fileSliceSize <- 2000
gene$LoadFile(expression_file_name)
## Load covariates
cvrt <- SlicedData$new()
cvrt$fileDelimiter <- "\t"
cvrt$fileOmitCharacters <- "NA"
cvrt$fileSkipRows <- 1
cvrt$fileSkipColumns <- 1
if (length(covariates_file_name) > 0) {
    cvrt$LoadFile(covariates_file_name)
}
## Run the analysis
snpspos <- read.table(snps_location_file_name, header = TRUE,
    stringsAsFactors = FALSE)
genepos <- read.table(gene_location_file_name, header = TRUE,
    stringsAsFactors = FALSE)
head(genepos)
me <- Matrix_eQTL_main(snps = snps, gene = gene, output_file_name =
    output_file_name_tra,
    pvOutputThreshold = pvOutputThreshold_tra, useModel = useModel,
    errorCovariance = numeric(), verbose = TRUE, output_file_name.cis =
        output_file_name_cis,
    pvOutputThreshold.cis = pvOutputThreshold_cis, snpspos = snpspos,
    genepos = genepos, cisDist = cisDist, pvalue.hist = TRUE, min.pv.by.
        genesnp = FALSE,
    noFDRsaveMemory = FALSE)
unlink(output_file_name_cis)
## Results:
cat("Analysis done in:", me$time.in.sec, " seconds", "\n")
cat("Detected local eQTLs:", "\n")
```

```
cis.eqtls <- me$cis$eqtls
cis.eqtls$beta_se <- cis.eqtls$beta/cis.eqtls$statistic
write.table(cis.eqtls, file = "sub.mouseliver.cis.1M.eqtls.txt", sep = "
    \t", row.names = FALSE, quote = FALSE)
### eqtl analysis for lung Settings Linear model to use, modelANOVA,
useModel <- modelLINEAR
# Genotype file name
SNP_file_name <- paste(base.dir, "/2016-09-08 BXD.geno.SNP.eqtl.for.lung
    .txt", sep = "")
snps_location_file_name <- paste(base.dir, "/2016-09-08 BXD.geno.loc.
    eqtl.for.lung.txt", sep = "")
# Gene expression file name
expression_file_name <- paste(base.dir, "/2016-09-08 mouse.lung.
    expression.eqtl.txt", sep = "")
gene_location_file_name <- paste(base.dir, "/2016-09-08 lung.gene.loc.
    txt", sep = "")
# Covariates file name Set to character() for no covariates
covariates_file_name <- character()
# Output file name
output_file_name_cis <- tempfile()
output_file_name_tra <- tempfile()
# Only associations significant at this level will be saved
pvOutputThreshold_cis <- 1
pvOutputThreshold_tra <- 5e-15
# Error covariance matrix Set to numeric() for identity.
errorCovariance <- numeric()
# errorCovariance = read.table('Sample_Data/errorCovariance.txt');
# Distance for local gene-SNP pairs
cisDist <- 1e+06
## Load genotype data
snps <- SlicedData$new()
snps$fileDelimiter <- "\t"
snps$fileOmitCharacters <- "NA"
```

```
snps$fileSkipRows <- 1
snps$fileSkipColumns <- 1
snps$fileSliceSize <- 2000
snps$LoadFile(SNP_file_name)
## Load gene expression data
gene <- SlicedData$new()
gene$fileDelimiter <- "\t"
gene$fileOmitCharacters <- "NA"
gene$fileSkipRows <- 1
gene$fileSkipColumns <- 1
gene$fileSliceSize <- 2000
gene$LoadFile(expression_file_name)
## Load covariates
cvrt <- SlicedData$new()
cvrt$fileDelimiter <- "\t"
cvrt$fileOmitCharacters <- "NA"
cvrt$fileSkipRows <- 1
cvrt$fileSkipColumns <- 1
if (length(covariates_file_name) > 0) {
    cvrt$LoadFile(covariates_file_name)
}
## Run the analysis
snpspos <- read.table(snps_location_file_name, header = TRUE,
    stringsAsFactors = FALSE)
genepos <- read.table(gene_location_file_name, header = TRUE,
    stringsAsFactors = FALSE)
head(genepos)
me <- Matrix_eQTL_main(snps = snps, gene = gene, output_file_name =
    output_file_name_tra,
    pvOutputThreshold = pvOutputThreshold_tra, useModel = useModel,
    errorCovariance = numeric(), verbose = TRUE, output_file_name.cis =
        output_file_name_cis,
    pvOutputThreshold.cis = pvOutputThreshold_cis, snpspos = snpspos,
```

```
141     genepos = genepos, cisDist = cisDist, pvalue.hist = TRUE, min.pv.by.
            genesnp = FALSE,
142     noFDRsaveMemory = FALSE)
143 unlink(output_file_name_cis)
144 ## Results:
145 cat("Analysis done in:", me$time.in.sec, " seconds", "\n")
146 cat("Detected local eQTLs:", "\n")
147 cis.eqtls <- me$cis$eqtls
148 cis.eqtls$beta_se <- cis.eqtls$beta/cis.eqtls$statistic
149 write.table(cis.eqtls, file = "mouselung.cis.1M.eqtls.txt", sep = "\t",
        row.names = FALSE, quote = FALSE)
```

B.3 Step 2 - Bayesian estimation

```
####### Bayesian Method load mouse lung cis eqtl result
lung.mouse.eQTL <- read.table(file = "mouselung.cis.1M.eqtls.txt",
    header = T)
# load mouse liver cis eqtl result
liver.mouse.eQTL <- read.table(file = "sub.mouseliver.cis.1M.eqtls.txt",
     header = T)
mouse4302ensembl_id <- read.table(file = "2015-12-04 mouse4302ensembl_id
    .txt", header = T)
mouse430aensembl_id <- read.table(file = "2015-12-07 mouse430aensembl_id
    .txt", header = T)
# Add ensemble id annoatation to the data
lung.mouse.eQTL <- merge(lung.mouse.eQTL, mouse4302ensembl_id, by.x = "
    gene", by.y = "probe_id")
liver.mouse.eQTL <- merge(liver.mouse.eQTL, mouse430aensembl_id, by.x =
    "gene", by.y = "probe_id")
library(data.table)
library(plyr)
# Select lung Gene-SNP pair with minimum P value
lung.mouse.eQTL.min <- data.table(lung.mouse.eQTL, key = c("ensembl_id",
     "pvalue"))
lung.mouse.eQTL.min <- lung.mouse.eQTL.min[J(unique(ensembl_id)), mult =
     "first"]
lung.mouse.eQTL.min <- as.data.frame(lung.mouse.eQTL.min)
# Select liver Gene-SNP pair with minimum P value
liver.mouse.eQTL.min <- data.table(liver.mouse.eQTL, key = c("ensembl_id
    ", "pvalue"))
liver.mouse.eQTL.min <- liver.mouse.eQTL.min[J(unique(ensembl_id)), mult
     = "first"]
liver.mouse.eQTL.min <- as.data.frame(liver.mouse.eQTL.min)
lung.mouse.eQTL.min <- rename(lung.mouse.eQTL.min, c(pvalue = "lung_
    pvalue", beta = "lung.beta", beta_se = "lung.beta_se"))
liver.mouse.eQTL.min <- rename(liver.mouse.eQTL.min, c(pvalue = "liver_
```

```
    pvalue", beta = "liver.beta", beta_se = "liver.beta_se"))
# lung, liver eqtl with ensemble_id
merged.mouse.eQTL.min <- merge(lung.mouse.eQTL.min, liver.mouse.eQTL.min
    , by.x = "ensembl_id", by.y = "ensembl_id")
head(merged.mouse.eQTL.min)
dim(merged.mouse.eQTL.min)
merged.mouse.eQTL.min <- data.frame(merged.mouse.eQTL.min)
merged.mouse.eQTL.min <- merged.mouse.eQTL.min[, c(1, 5, 7, 8, 12, 14,
    15)]
head(merged.mouse.eQTL.min)
write.table(merged.mouse.eQTL.min, file = "mouse.liver.expression.min.
    txt",
     sep = "\t", row.names = FALSE, quote = FALSE)
Pthreshold <- c(0.05, 0.01, 0.001, 1e-04, 1e-05, 1e-06, 1e-07, 1e-08, 1e
    -09)
eqtl.results <- matrix(0, nrow = length(Pthreshold), ncol = 5)
colnames(eqtl.results) <- c("Pvalue_threshold", "Sig_in_lung", "Percent_
    in_lung", "Sig_in_liver", "Percent_in_liver")
# Populate the said matrix
for (i in 1:length(Pthreshold)) {
    eqtl.results[i, 1] <- Pthreshold[i]
    eqtl.results[i, 2] <- sum(merged.mouse.eQTL.min$lung_pvalue <
        Pthreshold[i])
    eqtl.results[i, 3] <- sum(merged.mouse.eQTL.min$lung_pvalue <
        Pthreshold[i])/nrow(merged.mouse.eQTL.min)
    eqtl.results[i, 4] <- sum(merged.mouse.eQTL.min$liver_pvalue <
        Pthreshold[i])
    eqtl.results[i, 5] <- sum(merged.mouse.eQTL.min$liver_pvalue <
        Pthreshold[i])/nrow(merged.mouse.eQTL.min)
}
eqtl.results <- as.data.frame(eqtl.results)
eqtl.results$Pvalue_threshold <- as.character(eqtl.results$Pvalue_
    threshold)
```

```
eqtl.results$Sig_in_lung <- as.character(eqtl.results$Sig_in_lung)
eqtl.results$Sig_in_liver <- as.character(eqtl.results$Sig_in_liver)
eqtl.results$Percent_in_lung <- round(eqtl.results$Percent_in_lung, 2)
eqtl.results$Percent_in_liver <- round(eqtl.results$Percent_in_liver, 2)
eqtltab <- xtable(eqtl.results)
print.xtable(eqtltab, type = "latex", include.rownames = FALSE, file = "
    eqtltab.tex", latex.environments = "center")
Pvalue <- c(seq(from = 0.95, to = 0.1, length.out = 18), 0.05, 0.01,
    0.001, 1e-04, 1e-05, 1e-06, 1e-07, 5e-08)
chisq.results <- matrix(0, nrow = length(Pvalue), ncol = 5)
colnames(chisq.results) <- c("Pvalue_threshold", "Pvalue_chisq.test", "
    observed_shared", "expected_shared", "Folddifference")
# Populate the said matrix
for (i in 1:length(Pvalue)) {
    chisq.results[i, 1] <- Pvalue[i]
    a <- table(merged.mouse.eQTL.min$lung_pvalue<Pvalue[i], merged.mouse
        .eQTL.min$liver_pvalue<Pvalue[i])[,2]
    b <- chisq.test(table(merged.mouse.eQTL.min$lung_pvalue<Pvalue[i],
        merged.mouse.eQTL.min$liver_pvalue<Pvalue[i]),correct=T)$
        expected[, 2]
    c <- cbind(a,b)
    chisq.results[i,2] <- chisq.test(c,correct=T)$p.value
    chisq.results[i, 3] <- table(merged.mouse.eQTL.min$lung_pvalue <
        Pvalue[i], merged.mouse.eQTL.min$liver_pvalue < Pvalue[i])[2, 2]
    chisq.results[i, 4] <- chisq.test(table(merged.mouse.eQTL.min$lung_
        pvalue <
        Pvalue[i], merged.mouse.eQTL.min$liver_pvalue < Pvalue[i]),
        correct = T)$expected[2, 2]
    chisq.results[i, 5] <- chisq.results[i, 3]/chisq.results[i, 4]
}
print(chisq.results)
chisq.results.df <- as.data.frame(chisq.results)
library(ggplot2)
```

```
ae <- chisq.results.df[, c(1, 3, 4)]
ae1 <- data.frame(melt(ae, id.vars = "Pvalue_threshold"))
ae1$Pvalue_threshold <- as.numeric(ae1$Pvalue_threshold)
actvsexp <- ggplot(ae1, aes(x = -log10(Pvalue_threshold), y = value,
    color = variable)) + geom_line() + labs(y = "Number of overlapping
    cis-eQTL",
     x = expression("-log"[10] ~ "(P value threshold)"))
# actvsexp1<- actvsexp + guides(fill=guide_legend(title=NULL))
actvsexp1 <- actvsexp + scale_colour_discrete(name = " ", breaks = c("
    observed_shared",
     "expected_shared"), labels = c("observed overlap", "expected overlap
        ")) +
     scale_shape_discrete(name = " ", breaks = c("observed_shared", "
        expected_shared"),
         labels = c("observed overlap", "expected overlap")) + geom_vline
             (xintercept = -log10(0.05),
     color = "red", linetype = "dotted") + theme(legend.position = c
        (0.65, 0.8), text = element_text(size=15))
chisqfc <- ggplot(chisq.results.df, aes(x = -log10(Pvalue_threshold),
     y = Folddifference)) + geom_point() + labs(y = "Ratio of Observed vs
        .Expected ",
     x = expression("-log"[10] ~ "(P value threshold)")) + geom_hline(
        yintercept = 1,
     color = "red", linetype = "dotted") + theme(text = element_text(size
        =15))
# Multiple plot function ggplot objects can be passed in ..., or to
# plotlist (as a list of ggplot objects) - cols: Number of columns
# in layout - layout: A matrix specifying the layout. If present,
# 'cols' is ignored.  If the layout is something like
# matrix(c(1,2,3,3), nrow=2, byrow=TRUE), then plot 1 will go in the
# upper left, 2 will go in the upper right, and 3 will go all the
# way across the bottom.
multiplot <- function(..., plotlist = NULL, file, cols = 1, layout =
```

```
    NULL) {
93   library(grid)
94   # Make a list from the ... arguments and plotlist
95   plots <- c(list(...), plotlist)
96   numPlots <- length(plots)
97   # If layout is NULL, then use 'cols' to determine layout
98   if (is.null(layout)) {
99       # Make the panel ncol: Number of columns of plots nrow: Number
         of
100      # rows needed, calculated from # of cols
101      layout <- matrix(seq(1, cols * ceiling(numPlots/cols)), ncol =
         cols,
102          nrow = ceiling(numPlots/cols))
103  }
104  if (numPlots == 1) {
105      print(plots[[1]])
106  } else {
107      # Set up the page
108      grid.newpage()
109      pushViewport(viewport(layout = grid.layout(nrow(layout), ncol(
         layout))))
110      # Make each plot, in the correct location
111      for (i in 1:numPlots) {
112          # Get the i,j matrix positions of the regions that contain
             this
113          # subplot
114          matchidx <- as.data.frame(which(layout == i, arr.ind = TRUE)
             )
115          print(plots[[i]], vp = viewport(layout.pos.row = matchidx$
             row,
116              layout.pos.col = matchidx$col))
117      }
118  }
```

```
}
pdf("actvsexp.pdf", width = 7, height = 4.5)
multiplot(actvsexp1, chisqfc, cols = 2)
dev.off()
chisq.results.df$Pvalue_threshold <- as.character(chisq.results.df$
    Pvalue_threshold)
chisqfctab <- xtable(chisq.results.df, digits = c(0, 0, 4, 0, 0, 2))
chisqfctab
print.xtable(chisqfctab, type = "latex", file = "chisqfctab.tex", latex.
    environments = "center")
####### Start here for Bayesian analysis
merged.mouse.eQTL.min <- read.table(file = "mouse.liver.expression.min.
    txt",
     header = T)
merged.mouse.eQTL.min$abs_liver.beta <- abs(merged.mouse.eQTL.min$liver.
    beta)
merged.mouse.eQTL.min$abs_lung.beta <- abs(merged.mouse.eQTL.min$lung.
    beta)
merged.mouse.eQTL.min$abs_liver.beta <- abs(merged.mouse.eQTL.min$liver.
    beta)
merged.mouse.eQTL.min$abs_lung.beta <- abs(merged.mouse.eQTL.min$lung.
    beta)
merged.mouse.eQTL.min$neg_log_lung_pvalue <- -log10(merged.mouse.eQTL.
    min$lung_pvalue)
merged.mouse.eQTL.min$neg_log_liver_pvalue <- -log10(merged.mouse.eQTL.
    min$liver_pvalue)
cor.test(merged.mouse.eQTL.min$abs_lung.beta, merged.mouse.eQTL.min$neg_
    log_lung_pvalue)
# Make a basic volcano plot
vocano1 <- with(merged.mouse.eQTL.min, plot(lung.beta, -log10(lung_
    pvalue),
     xlab = expression(beta[lgm]), ylab = expression("-log"[10] ~ "(lung
         P value)"),
```

```
140     xlim = c(-4, 4), ylim = c(0, 40)))
141 vocano2 <- with(merged.mouse.eQTL.min, plot(liver.beta, -log10(liver_
      pvalue),
142     xlab = expression(beta[vgm]), ylab = expression("-log"[10] ~ "(liver
          P value)")))
143 pdf("volcano.pdf", width = 8, height = 6)
144 par(mfrow = c(1, 2))
145 par(mar=c(5,5,2,2))
146 with(merged.mouse.eQTL.min, plot(lung.beta, -log10(lung_pvalue), xlab =
      expression(beta[lgm]),
147     ylab = expression("-log"[10] ~ "(lung P value)"), cex.lab= 2.2, xlim
          = c(-4, 4), ylim = c(0, 40)))
148 with(merged.mouse.eQTL.min, plot(liver.beta, -log10(liver_pvalue), xlab
      = expression(beta[vgm]),
149     ylab = expression("-log"[10] ~ "(liver P value)"), cex.lab= 2.2,xlim
          = c(-4, 4), ylim = c(0, 40)))
150 dev.off()
151 cor(merged.mouse.eQTL.min$abs_lung.beta, merged.mouse.eQTL.min$neg_log_
      lung_pvalue)
152 ggplot(merged.mouse.eQTL.min, aes(x = abs_lung.beta, y = abs_liver.beta)
      ) + geom_point() +
153     xlab(expression(abs(hat(beta)) ~ "of lung")) + ylab(expression(abs(
          hat(beta)) ~ "of liver")) +
154     theme(text = element_text(size = 20)) + geom_abline(intercept = 0,
          slope = 1, colour = "red")


          tilde(beta))) + theme(text = element_text(size = 20)) + geom_
          abline(intercept = 0, slope = 1, colour = "red")
155 ggplot(merged.mouse.eQTL.min, aes(x = lung.beta, y = liver.beta)) + geom
      _point() +
156   xlab(expression(abs(hat(beta)) ~ "of lung")) + ylab(expression(abs(hat
        (beta)) ~ "of liver")) +
157   theme(text = element_text(size = 20)) + geom_abline(intercept = 0,
```

```
        slope = 1, colour = "red")
158
159 merged.mouse.eQTL <- merged.mouse.eQTL.min
160 # retrieve ensembl_id
161 markers <- merged.mouse.eQTL[, 1]
162 # Yg=Ag + Bg*Xsnp+V retrieve betas.hat (liver.beta)
163 betas.hat <- merged.mouse.eQTL$abs_liver.beta
164 # retrieve liver.beta_se
165 se <- merged.mouse.eQTL$liver.beta_se
166 # create Z matrix with 2 columns: 1 for intercept,abs_lung.beta
167 # (merged.mouse.eQTL[,10])
168 Z <- as.matrix(merged.mouse.eQTL$abs_lung.beta)
169 Z <- as.matrix(merged.mouse.eQTL$neg_log_lung_pvalue)  ##Use p-value as
      Z - didn't make a big difference
170 Z <- replace(Z, is.na(Z), 0)
171 Z <- data.frame(1, Z)
172 Z <- as.matrix(Z)
173 rowLength <- length(markers)
174 # Regression: abs_liver.beta = intercept + beta*abs_lung.beta +
175 # error
176 lmsummary <- summary(lm(abs_liver.beta ~ -1 + Z, data = merged.mouse.
      eQTL))
177 lmsummary
178 model.prior <- lm(abs_liver.beta ~ -1 + Z, data = merged.mouse.eQTL)
179 # error ~ N(0, Tau)
180 tau <- lmsummary$sigma^2
181 # output coeffieients (gamma matrix) gamma matrix
182 gamma <- as.matrix(lmsummary$coefficients[, 1])
183 # transpose Z matrix
184 Z_transpose <- t(Z)
185 # create identity matrix
186 identity <- diag(nrow = rowLength)
187 # original betas.hat
```

```
betas.hat <- as.matrix(betas.hat)
#### WEIGHTS
useweights <- 0  ##CHANGE TOGGLE
if (useweights == 1) {
    val <- 1
    weight <- exp(-merged.mouse.eQTL.min$neg_log_lung_pvalue + val)
}
# create V matrix for liver_residual_variance
V <- matrix(0, rowLength, rowLength)
# V, liver residual variance
diag(V) <- merged.mouse.eQTL$liver.beta_se^2
# Creat Tau matrix
Tau <- diag(tau, rowLength, rowLength)
# follow Chen's paper and caculate s
s <- V + Tau
if (useweights == 1) {
    s <- V + diag(weight) * Tau
}
# create inverse function for inversing diagnoal matrix
diag.inverse <- function(x) {
    diag(1/diag(x), nrow(x), ncol(x))
}
# create multiplication function for multiplicating two diagnoal
# matrix
diag.multi <- function(x, y) {
    diag(diag(x) * diag(y), nrow(x), ncol(x))
}
# inverse s
S <- diag.inverse(s)
# follow chen's paper to caculate omega
omega <- diag.multi(S, V)
# retrieve omega value from the matrix
omega.diag <- diag(omega)
```

```
# summary the omega value
summary(omega.diag)
# regression beta
regbeta <- Z %*% gamma
summary(regbeta)
betas.tieda0 <- omega %*% Z %*% gamma + (identity - omega) %*% betas.hat
markers1 <- as.character(markers)
# combine ensemble_id, betas.hat and betas.tieda
outputVector <- c(markers1, betas.hat, betas.tieda0, regbeta)
write.table(matrix(outputVector, rowLength), file = "hm_tau_hmresults0.
    txt", col.names = FALSE, row.names = FALSE, quote = FALSE)
liver.mouse.eQTL.bayesian <- read.table(file = "hm_tau_hmresults0.txt")
colnames(liver.mouse.eQTL.bayesian) <- c("ensembl_id", "betas.hat", "
    betas.tieda", "regbeta")
head(liver.mouse.eQTL.bayesian)
# merge dataset with betas.hat and betas.tieda
liver.mouse.eQTL.bayesian <- merge(liver.mouse.eQTL.bayesian, merged.
    mouse.eQTL.min, by = "ensembl_id")
head(liver.mouse.eQTL.bayesian)
write.table(liver.mouse.eQTL.bayesian, file = "liver.mouse.eQTL.
    bayesian0.txt")
# plotting for Comparison of beta and posterior estimations in
# unweighted Bayesian model
unweighted <- ggplot(liver.mouse.eQTL.bayesian, aes(x = betas.hat, y =
    betas.tieda)) + geom_point() + xlab(expression(abs(hat(beta)))) +
    ylab(expression("unweighted " ~
     tilde(beta))) + theme(text = element_text(size = 40)) + geom_abline(
         intercept = 0, slope = 1, colour = "red")
pdf("unweighted.pdf")
print(unweighted)
dev.off()
# caculate betas.tieda with the formula in Chen's paper
constant <- max(merged.mouse.eQTL.min$abs_liver.beta)/max(regbeta)  ###
```

```
    CHANGE
betas.tieda <- constant * omega %*% Z %*% gamma + (identity - omega) %*%
     betas.hat
markers1 <- as.character(markers)
# combine ensemble_id, betas.hat and betas.tieda
outputVector <- c(markers1, betas.hat, betas.tieda, regbeta)
write.table(matrix(outputVector, rowLength), file = "hm_tau_hmresults.
    txt", col.names = FALSE, row.names = FALSE, quote = FALSE)
liver.mouse.eQTL.bayesian <- read.table(file = "hm_tau_hmresults.txt")
colnames(liver.mouse.eQTL.bayesian) <- c("ensembl_id", "betas.hat", "
    betas.tieda", "regbeta")
head(liver.mouse.eQTL.bayesian)
# merge dataset with betas.hat and betas.tieda
liver.mouse.eQTL.bayesian <- merge(liver.mouse.eQTL.bayesian, merged.
    mouse.eQTL.min, by = "ensembl_id")
write.table(liver.mouse.eQTL.bayesian, file = "liver.mouse.eQTL.bayesian
    .txt")
# plotting for Comparison of beta and posterior estimations in
# weighted Bayesian model
weighted <- ggplot(liver.mouse.eQTL.bayesian, aes(x = betas.hat, y =
    betas.tieda)) + geom_point() + xlab(expression(abs(hat(beta)))) +
    ylab(expression("weighted " ~
     tilde(beta))) + theme(text = element_text(size = 40)) + geom_abline(
         intercept = 0, slope = 1, colour = "red")
pdf("weighted.pdf")
print(weighted)
dev.off()
pdf("betacompa.pdf")
multiplot(unweighted, weighted, cols = 2)
dev.off()
```

B.4 Step 3 - Posterior estimation

```
liver.mouse.eQTL.bayesian <- read.table(file = "liver.mouse.eQTL.
    bayesian.txt")
# Caculate variance for beta.tieda by following Brian Kulis' lecture
# notes Invert Tau and V
Tau_invert <- diag.inverse(Tau)
V_invert <- diag.inverse(V)
PS_invert <- Tau_invert + V_invert
# S in Brian Kulis' lecture note:PS
PS <- diag.inverse(PS_invert)
# retrieve posterior variance
ps <- diag(PS)
# reshape posterior variance to long format
ps.long <- melt(ps)
# Caculate sd: square root on variance
ps.long$betas.tieda.se <- (ps.long$value)^0.5
# combine sd to the data.frame
liver.mouse.eQTL.bayesian <- cbind(liver.mouse.eQTL.bayesian, ps.long$
    betas.tieda.se)
# head(liver.mouse.eQTL.bayesian) rename betas.tieda.se
liver.mouse.eQTL.bayesian <- rename(liver.mouse.eQTL.bayesian, c(`ps.
    long$betas.tieda.se` = "betas.tieda.se", liver.beta_se = "betas.hat.
    se"))
# caculate probability of betas.tieda below 0 based on betas.tieda
# and standard deviation
liver.mouse.eQTL.bayesian$p.below.0 <- pnorm(0, liver.mouse.eQTL.
    bayesian$betas.tieda, liver.mouse.eQTL.bayesian$betas.tieda.se)
pdf("boxplotpb0.pdf")
boxplot(liver.mouse.eQTL.bayesian$p.below.0, ylab = "value", xlab = "
    Probability below 0")
dev.off()
write.table(liver.mouse.eQTL.bayesian, file = "liver.mouse.eQTL.bayesian
     with beta.txt")
```

```
Bayesianbetasd <- basicStats(liver.mouse.eQTL.bayesian[, c(3, 15, 16)])[
    c("Mean", "Stdev", "Median", "Minimum", "Maximum"), ]
Bayesianbetasd <- xtable(Bayesianbetasd)
print.xtable(Bayesianbetasd, type = "latex", file = "Bayesianbetasd.tex"
    ,
    latex.environments = "center")
# Summary of posterior probability from weighted Bayesian method
eqtl.results1 <- matrix(0, nrow = length(Pthreshold), ncol = 3)
colnames(eqtl.results1) <- c("Pvalue_threshold", "Sig_in_liver", "
    Percent_in_liver")
for (i in 1:length(Pthreshold)) {
    eqtl.results1[i, 1] <- Pthreshold[i]
    eqtl.results1[i, 2] <- sum(liver.mouse.eQTL.bayesian$p.below.0 <
        Pthreshold[i])
    eqtl.results1[i, 3] <- sum(liver.mouse.eQTL.bayesian$p.below.0 <
        Pthreshold[i])/nrow(liver.mouse.eQTL.bayesian)
}
eqtl.results1 <- as.data.frame(eqtl.results1)
eqtl.results1$Pvalue_threshold <- as.character(eqtl.results1$Pvalue_
    threshold)
eqtl.results1$Sig_in_liver <- as.character(eqtl.results1$Sig_in_liver)
eqtl.results1$Percent_in_liver <- round(eqtl.results1$Percent_in_liver,
    2)
weqtltab <- xtable(eqtl.results1)
print.xtable(weqtltab, type = "latex", include.rownames = FALSE, file =
    "weqtltab.tex",
    latex.environments = "center")
```

B.5 Step 4 - Allele Specific Expression (ASE)

```
### START HERE
liver.mouse.eQTL.bayesian <- read.table(file = "liver.mouse.eQTL.
    bayesian with beta.txt")
liver.mouse.eQTL.bayesian.tau <- liver.mouse.eQTL.bayesian
### ASE
liver.ASE <- read.csv(file = "ASE.genetics.113.153882-6.csv")
# 440 unique gene ID
length(unique(liver.ASE$geneID))
# verify ASE table
liver.ASE1 <- liver.ASE[which(liver.ASE$replicate == "M.CH. DxB and BxD"
    ), ]
liver.ASE2 <- liver.ASE[which(liver.ASE$replicate == "M.HF DxB and BxD")
    , ]
liver.ASE3 <- liver.ASE[which(liver.ASE$replicate == "F.HF DxB and BxD")
    , ]
length(unique(liver.ASE1$geneID))
length(unique(liver.ASE2$geneID))
length(unique(liver.ASE3$geneID))
(length(unique(liver.ASE1$geneID)) + length(unique(liver.ASE2$geneID)) +
     length(unique(liver.ASE3$geneID)))/3
# As claimed in the paper: averaged 284 ASE for each replicate
sub.liver.ASE <- liver.ASE1
summary(sub.liver.ASE$pvalBH.DxB7)
sub.liver.ASE1 <- subset(sub.liver.ASE, pvalBH.DxB7 < 1e-14)
sub.liver.ASE2 <- subset(sub.liver.ASE, pvalBH.DxB7 >= 1e-14 & pvalBH.
    DxB7 < 5.8e-06)
sub.liver.ASE3 <- subset(sub.liver.ASE, pvalBH.DxB7 >= 5.8e-06 & pvalBH.
    DxB7 < 0.0031)
sub.liver.ASE4 <- subset(sub.liver.ASE, pvalBH.DxB7 >= 0.0031 & pvalBH.
    DxB7 >= 0.0031)
dim(sub.liver.ASE1)
dim(sub.liver.ASE2)
```

```
dim(sub.liver.ASE3)
dim(sub.liver.ASE4)
# sub.liver.ASE <- sub.liver.ASE[ sub.liver.ASE$geneID %in%
# names(table(sub.liver.ASE$geneID))[table(sub.liver.ASE$geneID) >1]
# , ] check the remain gene number after subsetting
dim(sub.liver.ASE)
liver.ASE.symbol <- unique(sub.liver.ASE$geneID)
liver.ASE.symbol1 <- unique(sub.liver.ASE1$geneID)
liver.ASE.symbol2 <- unique(sub.liver.ASE2$geneID)
liver.ASE.symbol3 <- unique(sub.liver.ASE3$geneID)
liver.ASE.symbol4 <- unique(sub.liver.ASE4$geneID)
length(liver.ASE.symbol)
# Annoate gene symbol with ensemble.ID
library(biomaRt)
mouse <- useMart("ensembl", dataset = "mmusculus_gene_ensembl")
liver.ASE.ensembl <- getBM(attributes = c("ensembl_gene_id", "mgi_symbol
    "), filters = "mgi_symbol", values = liver.ASE.symbol, mart = mouse)
liver.ASE.ensembl1 <- getBM(attributes = c("ensembl_gene_id", "mgi_
    symbol"), filters = "mgi_symbol", values = liver.ASE.symbol1, mart =
     mouse)
liver.ASE.ensembl2 <- getBM(attributes = c("ensembl_gene_id", "mgi_
    symbol"), filters = "mgi_symbol", values = liver.ASE.symbol2, mart =
     mouse)
liver.ASE.ensembl3 <- getBM(attributes = c("ensembl_gene_id", "mgi_
    symbol"), filters = "mgi_symbol", values = liver.ASE.symbol3, mart =
     mouse)
liver.ASE.ensembl4 <- getBM(attributes = c("ensembl_gene_id", "mgi_
    symbol"), filters = "mgi_symbol", values = liver.ASE.symbol4, mart =
     mouse)
dim(liver.ASE.ensembl)
liver.ASE.ensembl <- unique(liver.ASE.ensembl)

# delete liver ASE ensemble ID which are not in the
```

```
# liver.mouse.eQTL.bayesian data frame
liver.ASE.ensembl <- liver.ASE.ensembl[liver.ASE.ensembl$ensembl_gene_id
    %in% liver.mouse.eQTL.bayesian.tau$ensembl_id, ]
write.table(liver.ASE.ensembl, "liver.ASE.ensembl.txt")
liver.mouse.eQTL.bayesian.tau$eqtl[liver.mouse.eQTL.bayesian.tau$ensembl
    _id %in% liver.ASE.ensembl$ensembl_gene_id] <- 1
liver.mouse.eQTL.bayesian.tau$eqtl[!liver.mouse.eQTL.bayesian.tau$
    ensembl_id %in% liver.ASE.ensembl$ensembl_gene_id] <- 0
write.table(liver.mouse.eQTL.bayesian.tau, "liver.mouse.eQTL.bayesian.
    tau.txt")
summary(liver.mouse.eQTL.bayesian.tau$eqtl)
liver.mouse.eQTL.bayesian.tau$neg_log_liver_pvalue <- -log10(liver.mouse
    .eQTL.bayesian.tau$liver_pvalue)
by(liver.mouse.eQTL.bayesian.tau[, c(1, 7, 9, 14)], liver.mouse.eQTL.
    bayesian.tau[, "eqtl"], summary)
library(ggplot2)
boxplot(neg_log_liver_pvalue ~ eqtl, data = liver.mouse.eQTL.bayesian.
    tau, main = "liver.mouse.eQTL", xlab = "group", ylab = "liver neg
    log p")
boxplot(neg_log_lung_pvalue ~ eqtl, data = liver.mouse.eQTL.bayesian.tau
    , main = "lung.mouse.eQTL", xlab = "group", ylab = "lung neg log p")
liver.mouse.eQTL.bayesian.tau$eqtl[liver.mouse.eQTL.bayesian.tau$ensembl
    _id %in% liver.ASE.ensembl$ensembl_gene_id] <- "ASE"
liver.mouse.eQTL.bayesian.tau$eqtl[!liver.mouse.eQTL.bayesian.tau$
    ensembl_id %in% liver.ASE.ensembl$ensembl_gene_id] <- "Non-ASE"
pdf("boxplot01.pdf")
boxplot(neg_log_liver_pvalue ~ eqtl, data = liver.mouse.eQTL.bayesian.
    tau, main = "liver.mouse.eQTL", xlab = "group", ylab = "liver neg
    log p")
dev.off()
# boxplot(neg_log_lung_pvalue ~
# eqtl,data=liver.mouse.eQTL.bayesian.tau, main='lung.mouse.eQTL',
# xlab=group', ylab='lung neg log p', ylim=c(0, 16))
```

```
pdf("boxplot02.pdf")
boxplot(neg_log_lung_pvalue ~ eqtl, data = liver.mouse.eQTL.bayesian.tau
    , main = "lung.mouse.eQTL", xlab = "group", ylab = "lung neg log p")
dev.off()
pdf("boxplot.pdf", width = 9, height = 6)
par(mfrow = c(1, 2))
par(mar=c(5,5,2,2))
boxplot(neg_log_lung_pvalue ~ eqtl, data = liver.mouse.eQTL.bayesian.tau
    , main = "lung mouse cis-eQTL", xlab = "group", ylab = "lung neg log
     p", cex.lab= 1.8, cex.axis=1.5, ylim = c(0, 40), asp = 0.5)
boxplot(neg_log_liver_pvalue ~ eqtl, data = liver.mouse.eQTL.bayesian.
    tau, main = "liver mouse cis-eQTL", xlab = "group", ylab = "liver
    neg log p",cex.lab= 1.8, cex.axis=1.5, ylim = c(0, 40), asp = 0.5)
dev.off()
```

B.6 Step 5 - ROC plot and AUC analysis

```
liver.mouse.eQTL.bayesian.tau <- read.table("liver.mouse.eQTL.bayesian.
    tau.txt")
# chi-square (Fisher, 1932, Lancaster, 1961)
Fcomb <- function(ps) {
    k <- length(ps)
    temp <- -2 * sum(log(ps))
    pchisq(temp, 2 * k, lower.tail = F)
}
# normal (Liptak, 1958, Stouffer 1949)
Ncomb <- function(ps) {
    k <- length(ps)
    z <- qnorm((1 - ps))
    Ts <- sum(z)/sqrt(k)  # sum(1-Phi^-1(1-p))/sqrt(k)
    pnorm(Ts, lower.tail = F)  #Same as 1-Phi
}
# META
metapval <- apply(cbind(liver.mouse.eQTL.bayesian.tau$lung_pvalue, liver
    .mouse.eQTL.bayesian.tau$liver_pvalue),
    1, Ncomb)
liver.mouse.eQTL.bayesian.tau$metapval <- metapval
# Multiple posterior prob by 2 CHANGE?
# liver.mouse.eQTL.bayesian.tau$p.below.0 =
# 2*liver.mouse.eQTL.bayesian.tau$p.below.0 MT Method
mtresults <- read.table(paste0("MTeQTLs_ASE_3c_", sebsetn, "s.txt"),
    header = TRUE)
minmtresults <- sapply(liver.mouse.eQTL.bayesian.tau$ensembl_id,
    function(x) min(mtresults[mtresults$ensembl_id == as.character(x), "
    marginalP.liver"]))
newmtresults <- data.frame(liver.mouse.eQTL.bayesian.tau$ensembl_id,
    minmtresults, liver.mouse.eQTL.bayesian.tau$eqtl)
colnames(newmtresults) <- c("ensembl_id", "marginalp", "eqtl")
newresults <- liver.mouse.eQTL.bayesian.tau[, c("ensembl_id", "lung_
```

```
    pvalue", "liver_pvalue", "metapval", "p.below.0", "eqtl")]
library(pROC)
# ROC plotting
pdf(paste0("subsampleproc1", sebsetn, ".pdf"), width = 4, height = 4)
rocobj1 <- plot.roc(newresults$eqtl, newresults$liver_pvalue, col = "
    black", legacy.axes = TRUE, yaxs = "i")
rocobj2 <- lines.roc(newresults$eqtl, newresults$p.below.0, col = "red")
rocobj3 <- lines.roc(newmtresults$eqtl, newmtresults$marginalp, col = "
    green")
rocobj4 <- lines.roc(newresults$eqtl, newresults$metapval, col = "blue")
rocobj5 <- lines.roc(newresults$eqtl, newresults$lung_pvalue, col = "
    purple")
legend("bottomright", legend = c("Conventional liver", "TA-eQTL", "MT",
    "Meta", "Conventional lung"), col = c("black", "red", "green", "blue
    ", "purple"), lwd = 2, cex = 0.75, bty = "n")
dev.off()

# ROC plotting
pdf(paste0("subsampleproc14", sebsetn, ".pdf"), width = 4, height = 4)
rocobj1 <- plot.roc(newresults$eqtl, newresults$liver_pvalue, col = "
    black", legacy.axes = TRUE, yaxs = "i")
rocobj2 <- lines.roc(newresults$eqtl, newresults$p.below.0, col = "red")
rocobj3 <- lines.roc(newmtresults$eqtl, newmtresults$marginalp, col = "
    green")
rocobj4 <- lines.roc(newresults$eqtl, newresults$metapval, col = "blue")
rocobj5 <- lines.roc(newresults$eqtl, newresults$lung_pvalue, col = "
    purple")
legend("bottomright", legend = c("Conventional liver", "TA-eQTL", "MT",
    "Meta", "Conventional lung"), col = c("black", "red", "green", "blue
    ", "purple"), lwd = 2, cex = 1.5, bty = "n")
dev.off()

orig_auc1 <- as.numeric(ci(newresults$eqtl, newresults$liver_pvalue))
```

```
bayesian_auc1 <- as.numeric(ci(newresults$eqtl, newresults$p.below.0))
lung_auc1 <- as.numeric(ci(newresults$eqtl, newresults$lung_pvalue))
meta_auc1 <- as.numeric(ci(newresults$eqtl, newresults$metapval))
mt_auc1 <- as.numeric(ci(newmtresults$eqtl, newmtresults$marginalp))
auc1 <- rbind(orig_auc1, bayesian_auc1, mt_auc1, meta_auc1, lung_auc1)
colnames(auc1) <- c("lowerCI", "mean", "upperCI")
auc1 <- data.frame(auc1)
auc2 <- round(auc1[, ], 2)
auc2$CI <- paste(auc2$lowerCI, auc2$upperCI, sep = ", ")
auc2$CI <- paste("(", auc2$CI, ")", sep = "")
auc2$lowerCI <- NULL
auc2$upperCI <- NULL
colnames(auc2) <- c("AUC", "CI")
rownames(auc2) <- c("Conventional liver", "TA-eQTL", "MT", "Meta", "
    Conventional lung")
auctable <- xtable(auc2)
print.xtable(auctable, type = "latex", file = paste0("auc", sebsetn, ".
    tex"), latex.environments = "center")
auc3 <- auc1
rownames(auc3) <- c("Conventional liver", "TA-eQTL", "MT", "Meta", "
    Conventional lung")
auc3$methods <- factor(row.names(auc3))
positions <- c("Conventional liver", "TA-eQTL", "MT", "Meta", "
    Conventional lung")
aucfivemethods <- ggplot(auc3, aes(x = methods, y = mean)) + geom_bar(
    stat = "identity",
    fill = c("black", "red", "green", "blue", "purple")) + xlab("Methods
        ") +
    ylab("Area under the curve (AUC)") + geom_errorbar(aes(ymin =
        lowerCI,
    ymax = upperCI), width = 0.1) + scale_x_discrete(limits = positions)
        +
    coord_cartesian(ylim = c(0.5, 0.9)) + theme(axis.title.y = element_
```

```
        text(size = rel(1.8),
74    angle = 90)) + theme(axis.title.x = element_text(size = rel(1.8),
        angle = 0)) +
75    theme(axis.text.x = element_text(face = "bold", size = 12))
76 pdf("aucfivemethods.pdf")
77 print(aucfivemethods)
78 dev.off()
79
80
81 # significant testing to compare two ROC curves
82 orig.roc <- roc(newresults$eqtl, newresults$liver_pvalue)
83 bayesian.roc <- roc(newresults$eqtl, newresults$p.below.0)
84 mt.roc <- roc(newmtresults$eqtl, newmtresults$marginalp)
85 meta.roc <- roc(newresults$eqtl, newresults$metapval)
86 roc.test(orig.roc, bayesian.roc)
87 roc.test(orig.roc, mt.roc)
88 roc.test(orig.roc, meta.roc)
89 roc.test(mt.roc, bayesian.roc)
```

B.7 Step 6 - Model performance assessment on subsetted samples

```
1 rm(list = ls())
2 gc()
3 # set directory
4 setwd("/Volumes/Transcend/Thesis_project/eQTL data")
5 library(pROC)
6 library("MatrixEQTL")
7 library(fBasics)
8 library(plyr)
9 library(xtable)
10 library(data.table)
11 library(biomaRt)
12 library(ggplot2)
```

```
library(lme4)
library(lsmeans)
options(xtable.floating = FALSE)
options(xtable.timestamp = "")
# subset dataset
combined_auc <- NULL
pdf("subsample.pdf", width = 8, height = 12)
par(mfrow = c(3, 2))
par(cex.lab= 2, cex.main=2)
# seed library
seedlib <- c(45:50)
aorig_auc <- abayesian_auc <- alung_auc <- ameta_auc <- amt_auc <- NULL
# set sub-sampling options: 10, 15, 20, 25, 30 strains
sublib <- c(25, 20, 15, 10, 30)
# loop to subsampling analyses
for (z in 1:length(sublib)) {
    sebsetn <- sublib[z]
    # full liver dataset has 30 strains
    combinded.results <- NULL
    orig_auc <- matrix(0, length(seedlib), 3)
    colnames(orig_auc) <- c("sebsetn", "samplingseed", "auc")
    bayesian_auc <- lung_auc <- meta_auc <- mt_auc <- orig_auc
    p <- 1
    for (k in 1:length(seedlib)) {
        set.seed(seedlib[k])
        # subset liver gene expression dataset
        mouse.liver.expression.eqtl <- read.table(file = "2016-09-08
            mouse.liver.expression.eqtl.txt",
            header = T)
        sub.mouse.liver.expression.eqtl <- mouse.liver.expression.eqtl[,
            c(1, sample(2:dim(mouse.liver.expression.eqtl)[2], sebsetn,
                replace = FALSE))]
        write.table(sub.mouse.liver.expression.eqtl, file = "sub.mouse.
```

```
    liver.expression.eqtl.txt",
45      sep = "\t", row.names = FALSE, quote = FALSE)
46  # subset liver snp expression data
47  BXD.geno.SNP.eqtl.for.liver <- read.table(file = "2016-09-08 BXD
    .geno.SNP.eqtl.for.liver.txt", header = T)
48  head(BXD.geno.SNP.eqtl.for.liver)
49  dim(BXD.geno.SNP.eqtl.for.liver)
50  set.seed(seedlib[k])
51  sub.BXD.geno.SNP.eqtl.for.liver <- BXD.geno.SNP.eqtl.for.liver[,
      c(1, sample(2:dim(BXD.geno.SNP.eqtl.for.liver)[2], sebsetn,
      replace = FALSE))]
52  head(sub.BXD.geno.SNP.eqtl.for.liver)
53  dim(sub.BXD.geno.SNP.eqtl.for.liver)
54  write.table(sub.BXD.geno.SNP.eqtl.for.liver, file = "sub.BXD.
    geno.SNP.eqtl.for.liver.txt", sep = "\t", row.names = FALSE,
      quote = FALSE)
55  ### MT eqtl analysis
56  source("2016-09-12mtsubsetanalysis.R")
57  ### liver eqtl analysis
58  base.dir <- "/Volumes/Transcend/Thesis_project/eQTL data"
59  # Linear model to use, modelANOVA, modelLINEAR, or modelLINEAR_
    CROSS
60  useModel <- modelLINEAR
61  # Genotype file name
62  SNP_file_name <- paste(base.dir, "/sub.BXD.geno.SNP.eqtl.for.
    liver.txt", sep = "")
63  snps_location_file_name <- paste(base.dir, "/2016-09-08 BXD.geno
    .loc.eqtl.for.liver.txt", sep = "")
64  # Gene expression file name
65  expression_file_name <- paste(base.dir, "/sub.mouse.liver.
    expression.eqtl.txt", sep = "")
66  gene_location_file_name <- paste(base.dir, "/2016-09-08 liver.
    gene.loc.txt", sep = "")
```

```
67        # Covariates file name Set to character() for no covariates
68        covariates_file_name <- character()
69        # Output file name
70        output_file_name_cis <- tempfile()
71        output_file_name_tra <- tempfile()
72        # Only associations significant at this level will be saved
73        pvOutputThreshold_cis <- 1
74        pvOutputThreshold_tra <- 5e-15
75        # Error covariance matrix Set to numeric() for identity.
76        errorCovariance <- numeric()
77        # errorCovariance = read.table('Sample_Data/errorCovariance.txt
              ');
78        # Distance for local gene-SNP pairs
79        cisDist <- 1e+06
80        ## Load genotype data
81        snps <- SlicedData$new()
82        snps$fileDelimiter <- "\t"
83        snps$fileOmitCharacters <- "NA"
84        snps$fileSkipRows <- 1
85        snps$fileSkipColumns <- 1
86        snps$fileSliceSize <- 2000
87        snps$LoadFile(SNP_file_name)
88        ## Load gene expression data
89        gene <- SlicedData$new()
90        gene$fileDelimiter <- "\t"
91        gene$fileOmitCharacters <- "NA"
92        gene$fileSkipRows <- 1
93        gene$fileSkipColumns <- 1
94        gene$fileSliceSize <- 2000
95        gene$LoadFile(expression_file_name)
96        ## Load covariates
97        cvrt <- SlicedData$new()
98        cvrt$fileDelimiter <- "\t"
```

```
99          cvrt$fileOmitCharacters <- "NA"
100         cvrt$fileSkipRows <- 1
101         cvrt$fileSkipColumns <- 1
102         if (length(covariates_file_name) > 0) {
103             cvrt$LoadFile(covariates_file_name)
104         }
105         ## Run the analysis
106         snpspos <- read.table(snps_location_file_name, header = TRUE,
                stringsAsFactors = FALSE)
107         genepos <- read.table(gene_location_file_name, header = TRUE,
                stringsAsFactors = FALSE)
108         head(genepos)
109         me <- Matrix_eQTL_main(snps = snps, gene = gene, output_file_
                name = output_file_name_tra,
110             pvOutputThreshold = pvOutputThreshold_tra, useModel =
                    useModel,
111             errorCovariance = numeric(), verbose = TRUE, output_file_
                    name.cis = output_file_name_cis,
112             pvOutputThreshold.cis = pvOutputThreshold_cis, snpspos =
                    snpspos,
113             genepos = genepos, cisDist = cisDist, pvalue.hist = TRUE,
                    min.pv.by.genesnp = FALSE, noFDRsaveMemory = FALSE)
114         unlink(output_file_name_cis)
115         ## Results:
116         cat("Analysis done in:", me$time.in.sec, " seconds", "\n")
117         cat("Detected local eQTLs:", "\n")
118         cis.eqtls <- me$cis$eqtls
119         head(cis.eqtls)
120         dim(cis.eqtls)
121         cis.eqtls$beta_se <- cis.eqtls$beta/cis.eqtls$statistic
122         write.table(cis.eqtls, file = "sub.mouseliver.cis.1M.eqtls.txt",
                sep = "\t", row.names = FALSE, quote = FALSE)
123         ### eqtl analysis for lung Settings Linear model to use,
```

```
    modelANOVA,
### modelLINEAR, or modelLINEAR_CROSS
useModel <- modelLINEAR
# Genotype file name
SNP_file_name <- paste(base.dir, "/2016-09-08 BXD.geno.SNP.eqtl.
    for.lung.txt", sep = "")
snps_location_file_name <- paste(base.dir, "/2016-09-08 BXD.geno
    .loc.eqtl.for.lung.txt", sep = "")
# Gene expression file name
expression_file_name <- paste(base.dir, "/2016-09-08 mouse.lung.
    expression.eqtl.txt", sep = "")
gene_location_file_name <- paste(base.dir, "/2016-09-08 lung.
    gene.loc.txt", sep = "")
# Covariates file name Set to character() for no covariates
covariates_file_name <- character()
# Output file name
output_file_name_cis <- tempfile()
output_file_name_tra <- tempfile()
# Only associations significant at this level will be saved
pvOutputThreshold_cis <- 1
pvOutputThreshold_tra <- 5e-15
# Error covariance matrix Set to numeric() for identity.
errorCovariance <- numeric()
# errorCovariance = read.table('Sample_Data/errorCovariance.txt
    ');
# Distance for local gene-SNP pairs
cisDist <- 1e+06
## Load genotype data
snps <- SlicedData$new()
snps$fileDelimiter <- "\t"
snps$fileOmitCharacters <- "NA"
snps$fileSkipRows <- 1
snps$fileSkipColumns <- 1
```

```
151         snps$fileSliceSize <- 2000
152         snps$LoadFile(SNP_file_name)
153         ## Load gene expression data
154         gene <- SlicedData$new()
155         gene$fileDelimiter <- "\t"
156         gene$fileOmitCharacters <- "NA"
157         gene$fileSkipRows <- 1
158         gene$fileSkipColumns <- 1
159         gene$fileSliceSize <- 2000
160         gene$LoadFile(expression_file_name)
161         ## Load covariates
162         cvrt <- SlicedData$new()
163         cvrt$fileDelimiter <- "\t"
164         cvrt$fileOmitCharacters <- "NA"
165         cvrt$fileSkipRows <- 1
166         cvrt$fileSkipColumns <- 1
167         if (length(covariates_file_name) > 0) {
168             cvrt$LoadFile(covariates_file_name)
169         }
170         ## Run the analysis
171         snpspos <- read.table(snps_location_file_name, header = TRUE,
              stringsAsFactors = FALSE)
172         genepos <- read.table(gene_location_file_name, header = TRUE,
              stringsAsFactors = FALSE)
173         head(genepos)
174         me <- Matrix_eQTL_main(snps = snps, gene = gene, output_file_
              name = output_file_name_tra,
175             pvOutputThreshold = pvOutputThreshold_tra, useModel =
                  useModel,
176             errorCovariance = numeric(), verbose = TRUE, output_file_
                  name.cis = output_file_name_cis,
177             pvOutputThreshold.cis = pvOutputThreshold_cis, snpspos =
                  snpspos,
```

```
178             genepos = genepos, cisDist = cisDist, pvalue.hist = TRUE,
                    min.pv.by.genesnp = FALSE,
179             noFDRsaveMemory = FALSE)
180         unlink(output_file_name_cis)
181         ## Results:
182         cis.eqtls <- me$cis$eqtls
183         head(cis.eqtls)
184         dim(cis.eqtls)
185         cis.eqtls$beta_se <- cis.eqtls$beta/cis.eqtls$statistic
186         write.table(cis.eqtls, file = "mouselung.cis.1M.eqtls.txt", sep
                = "\t", row.names = FALSE, quote = FALSE)
187         ####### Bayesian Method load mouse lung cis eqtl result
188         lung.mouse.eQTL <- read.table(file = "mouselung.cis.1M.eqtls.txt
                ", header = T)
189         # load mouse liver cis eqtl result
190         liver.mouse.eQTL <- read.table(file = "sub.mouseliver.cis.1M.
                eqtls.txt", header = T)
191         mouse4302ensembl_id <- read.table(file = "2015-12-04
                mouse4302ensembl_id.txt", header = T)
192         mouse430aensembl_id <- read.table(file = "2015-12-07
                mouse430aensembl_id.txt", header = T)
193         # Add ensemble id annoatation to the data
194         lung.mouse.eQTL <- merge(lung.mouse.eQTL, mouse4302ensembl_id,
                by.x = "gene", by.y = "probe_id")
195         liver.mouse.eQTL <- merge(liver.mouse.eQTL, mouse430aensembl_id,
                 by.x = "gene", by.y = "probe_id")
196         # Select lung Gene-SNP pair with minimum P value
197         lung.mouse.eQTL.min <- data.table(lung.mouse.eQTL, key = c("
                ensembl_id", "pvalue"))
198         lung.mouse.eQTL.min <- lung.mouse.eQTL.min[J(unique(ensembl_id))
                , mult = "first"]
199         lung.mouse.eQTL.min <- as.data.frame(lung.mouse.eQTL.min)
200         # Select liver Gene-SNP pair with minimum P value
```

```
liver.mouse.eQTL.min <- data.table(liver.mouse.eQTL, key = c("
    ensembl_id", "pvalue"))
liver.mouse.eQTL.min <- liver.mouse.eQTL.min[J(unique(ensembl_id
    )), mult = "first"]
liver.mouse.eQTL.min <- as.data.frame(liver.mouse.eQTL.min)
lung.mouse.eQTL.min <- rename(lung.mouse.eQTL.min, c(pvalue = "
    lung_pvalue", beta = "lung.beta", beta_se = "lung.beta_se"))
liver.mouse.eQTL.min <- rename(liver.mouse.eQTL.min, c(pvalue =
    "liver_pvalue", beta = "liver.beta", beta_se = "liver.beta_
    se"))
# lung, liver eqtl with ensemble_id
merged.mouse.eQTL.min <- merge(lung.mouse.eQTL.min, liver.mouse.
    eQTL.min, by.x = "ensembl_id", by.y = "ensembl_id")
merged.mouse.eQTL.min <- data.frame(merged.mouse.eQTL.min)
merged.mouse.eQTL.min <- merged.mouse.eQTL.min[, c(1, 5, 7, 8,
    12, 14, 15)]
head(merged.mouse.eQTL.min)
write.table(merged.mouse.eQTL.min, file = "mouse.liver.
    expression.min.txt", sep = "\t", row.names = FALSE, quote =
    FALSE)
####### START HERE
merged.mouse.eQTL.min <- read.table(file = "mouse.liver.
    expression.min.txt", header = T)
merged.mouse.eQTL.min$abs_liver.beta <- abs(merged.mouse.eQTL.
    min$liver.beta)
merged.mouse.eQTL.min$abs_lung.beta <- abs(merged.mouse.eQTL.min
    $lung.beta)
merged.mouse.eQTL.min$abs_liver.beta <- abs(merged.mouse.eQTL.
    min$liver.beta)
merged.mouse.eQTL.min$abs_lung.beta <- abs(merged.mouse.eQTL.min
    $lung.beta)
merged.mouse.eQTL.min$neg_log_lung_pvalue <- -log10(merged.mouse
    .eQTL.min$lung_pvalue)
```

```
219         merged.mouse.eQTL.min$neg_log_liver_pvalue <- -log10(merged.
               mouse.eQTL.min$liver_pvalue)
220         merged.mouse.eQTL <- merged.mouse.eQTL.min
221         # retrieve ensembl_id
222         markers <- merged.mouse.eQTL[, 1]
223         # Yg=Ag + Bg*Xsnp+V retrieve betas.hat (liver.beta)
224         betas.hat <- merged.mouse.eQTL$abs_liver.beta
225         # retrieve liver.beta_se
226         se <- merged.mouse.eQTL$liver.beta_se
227         # create Z matrix with 2 columns: 1 for intercept,abs_lung.beta
228         # (merged.mouse.eQTL[,10])
229         Z <- as.matrix(merged.mouse.eQTL$abs_lung.beta)
230         Z <- as.matrix(merged.mouse.eQTL$neg_log_lung_pvalue)  ##Use p-
               value as Z - didn't make a big difference
231         Z <- replace(Z, is.na(Z), 0)
232         Z <- data.frame(1, Z)
233         Z <- as.matrix(Z)
234         rowLength <- length(markers)
235         # Regression: abs_liver.beta = intercept + beta*abs_lung.beta +
               error
236         lmsummary <- summary(lm(abs_liver.beta ~ -1 + Z, data = merged.
               mouse.eQTL))
237         lmsummary
238         model.prior <- lm(abs_liver.beta ~ -1 + Z, data = merged.mouse.
               eQTL)
239         # error ~ N(0, Tau)
240         tau <- lmsummary$sigma^2
241         tau
242         # output coeffieients (gamma matrix) gamma matrix
243         gamma <- as.matrix(lmsummary$coefficients[, 1])
244         # transpose Z matrix
245         Z_transpose <- t(Z)
246         # create identity matrix
```

```
247         identity <- diag(nrow = rowLength)
248         # original betas.hat
249         betas.hat <- as.matrix(betas.hat)
250         #### WEIGHTS
251         useweights <- 0  ##CHANGE TOGGLE
252         if (useweights == 1) {
253             val <- 1
254             weight <- exp(-merged.mouse.eQTL.min$neg_log_lung_pvalue +
255                 val)
256         }
257         # create V matrix for liver_residual_variance
258         V <- matrix(0, rowLength, rowLength)
259         # V, liver residual variance
260         diag(V) <- merged.mouse.eQTL$liver.beta_se^2
261         # Creat Tau matrix
262         Tau <- diag(tau, rowLength, rowLength)
263         # follow Chen's paper and caculate s
264         s <- V + Tau
265         if (useweights == 1) {
266             s <- V + diag(weight) * Tau
267         }
268         # create inverse function for inversing diagnoal matrix
269         diag.inverse <- function(x) {
270             diag(1/diag(x), nrow(x), ncol(x))
271         }
272         # create multiplication function for multiplicating two diagnoal
                matrix
273         diag.multi <- function(x, y) {
274             diag(diag(x) * diag(y), nrow(x), ncol(x))
275         }
276         # inverse s
277         S <- diag.inverse(s)
278         # follow chen's paper to caculate omega
```

```
279         omega <- diag.multi(S, V)
280         # retrieve omega value from the matrix
281         omega.diag <- diag(omega)
282         # summary the omega value
283         summary(omega.diag)
284         # regression beta
285         regbeta <- Z %*% gamma
286         summary(regbeta)
287         betas.tieda0 <- omega %*% Z %*% gamma + (identity - omega) %*%
288             betas.hat
289         markers1 <- as.character(markers)
290         # combine ensemble_id, betas.hat and betas.tieda
291         outputVector <- c(markers1, betas.hat, betas.tieda0, regbeta)
292         write.table(matrix(outputVector, rowLength), file = "hm_tau_
                hmresults0.txt", col.names = FALSE, row.names = FALSE, quote
                 = FALSE)
293         liver.mouse.eQTL.bayesian <- read.table(file = "hm_tau_
                hmresults0.txt")
294         colnames(liver.mouse.eQTL.bayesian) <- c("ensembl_id", "betas.
                hat", "betas.tieda", "regbeta")
295         head(liver.mouse.eQTL.bayesian)
296         # merge dataset with betas.hat and betas.tieda
297         liver.mouse.eQTL.bayesian <- merge(liver.mouse.eQTL.bayesian,
                merged.mouse.eQTL.min, by = "ensembl_id")
298         write.table(liver.mouse.eQTL.bayesian, file = "liver.mouse.eQTL.
                bayesian0.txt")
299         # caculate betas.tieda with the formula in Chen's paper
300         constant <- max(merged.mouse.eQTL.min$abs_liver.beta)/max(
                regbeta)  ###CHANGE
301         betas.tieda <- constant * omega %*% Z %*% gamma + (identity -
                omega) %*% betas.hat
302         markers1 <- as.character(markers)
303         # combine ensemble_id, betas.hat and betas.tieda
```

```
304         outputVector <- c(markers1, betas.hat, betas.tieda, regbeta)
305         write.table(matrix(outputVector, rowLength), file = "hm_tau_
                hmresults.txt", col.names = FALSE, row.names = FALSE, quote
                = FALSE)
306         liver.mouse.eQTL.bayesian <- read.table(file = "hm_tau_hmresults
                .txt")
307         colnames(liver.mouse.eQTL.bayesian) <- c("ensembl_id", "betas.
                hat", "betas.tieda", "regbeta")
308         # merge dataset with betas.hat and betas.tieda
309         liver.mouse.eQTL.bayesian <- merge(liver.mouse.eQTL.bayesian,
                merged.mouse.eQTL.min, by = "ensembl_id")
310         write.table(liver.mouse.eQTL.bayesian, file = "liver.mouse.eQTL.
                bayesian.txt")
311         liver.mouse.eQTL.bayesian <- read.table(file = "liver.mouse.eQTL
                .bayesian.txt")
312         # Caculate variance for beta.tieda by following Brian Kulis'
                lecture
313         # notes Invert Tau and V
314         Tau_invert <- diag.inverse(Tau)
315         V_invert <- diag.inverse(V)
316         PS_invert <- Tau_invert + V_invert
317         # S in Brian Kulis' lecture note:PS
318         PS <- diag.inverse(PS_invert)
319         # retrieve posterior variance
320         ps <- diag(PS)
321         # reshape posterior variance to long format
322         ps.long <- melt(ps)
323         # Caculate sd: square root on variance
324         ps.long$betas.tieda.se <- (ps.long$value)^0.5
325         # combine sd to the data.frame
326         liver.mouse.eQTL.bayesian <- cbind(liver.mouse.eQTL.bayesian, ps
                .long$betas.tieda.se)
327         # rename betas.tieda.se
```

```
liver.mouse.eQTL.bayesian <- rename(liver.mouse.eQTL.bayesian, c
    (`ps.long$betas.tieda.se` = "betas.tieda.se", liver.beta_se
    = "betas.hat.se"))
liver.mouse.eQTL.bayesian$p.below.0 <- pnorm(0, liver.mouse.eQTL
    .bayesian$betas.tieda, liver.mouse.eQTL.bayesian$betas.tieda
    .se)
write.table(liver.mouse.eQTL.bayesian, file = "liver.mouse.eQTL.
    bayesian with beta.txt")
### START HERE
liver.mouse.eQTL.bayesian <- read.table(file = "liver.mouse.eQTL
    .bayesian with beta.txt")
liver.mouse.eQTL.bayesian.tau <- liver.mouse.eQTL.bayesian
### ASE
liver.ASE <- read.csv(file = "ASE.genetics.113.153882-6.csv")
# 440 unique gene ID
length(unique(liver.ASE$geneID))
# verify ASE table
liver.ASE1 <- liver.ASE[which(liver.ASE$replicate == "M.CH. DxB
    and BxD"), ]
sub.liver.ASE <- liver.ASE1
summary(sub.liver.ASE$pvalBH.DxB7)
# sub.liver.ASE <- sub.liver.ASE[ sub.liver.ASE$geneID %in%
# names(table(sub.liver.ASE$geneID))[table(sub.liver.ASE$geneID)
     >1] ,
# ] check the remain gene number after subsetting
liver.ASE.symbol <- unique(sub.liver.ASE$geneID)
# Annoate gene symbol with ensemble.ID
mouse <- useMart("ensembl", dataset = "mmusculus_gene_ensembl")
liver.ASE.ensembl <- getBM(attributes = c("ensembl_gene_id", "
    mgi_symbol"),
     filters = "mgi_symbol", values = liver.ASE.symbol, mart =
         mouse)
liver.ASE.ensembl <- unique(liver.ASE.ensembl)
```

```
# delete liver ASE ensemble ID which are not in the
# liver.mouse.eQTL.bayesian data frame
liver.ASE.ensembl <- liver.ASE.ensembl[liver.ASE.ensembl$ensembl
    _gene_id %in% liver.mouse.eQTL.bayesian.tau$ensembl_id, ]
liver.mouse.eQTL.bayesian.tau$eqtl[liver.mouse.eQTL.bayesian.tau
    $ensembl_id %in% liver.ASE.ensembl$ensembl_gene_id] <- 1
liver.mouse.eQTL.bayesian.tau$eqtl[!liver.mouse.eQTL.bayesian.
    tau$ensembl_id %in% liver.ASE.ensembl$ensembl_gene_id] <- 0
write.table(liver.mouse.eQTL.bayesian.tau, "liver.mouse.eQTL.
    bayesian.tau.txt")
liver.mouse.eQTL.bayesian.tau$neg_log_liver_pvalue <- -log10(
    liver.mouse.eQTL.bayesian.tau$liver_pvalue)
by(liver.mouse.eQTL.bayesian.tau[, c(1, 7, 9, 14)], liver.mouse.
    eQTL.bayesian.tau[, "eqtl"], summary)
liver.mouse.eQTL.bayesian.tau$eqtl[liver.mouse.eQTL.bayesian.tau
    $ensembl_id %in% liver.ASE.ensembl$ensembl_gene_id] <- 1
liver.mouse.eQTL.bayesian.tau$eqtl[!liver.mouse.eQTL.bayesian.
    tau$ensembl_id %in% liver.ASE.ensembl$ensembl_gene_id] <- 0
liver.mouse.eQTL.bayesian.tau <- read.table("liver.mouse.eQTL.
    bayesian.tau.txt")
# chi-square (Fisher, 1932, Lancaster, 1961)
Fcomb <- function(ps) {
    k <- length(ps)
    temp <- -2 * sum(log(ps))
    pchisq(temp, 2 * k, lower.tail = F)
}
# normal (Liptak, 1958, Stouffer 1949)
Ncomb <- function(ps) {
    k <- length(ps)
    z <- qnorm((1 - ps))
    Ts <- sum(z)/sqrt(k)  # sum(1-Phi^-1(1-p))/sqrt(k)
    pnorm(Ts, lower.tail = F)  #Same as 1-Phi
}
```

```
375         # META
376         metapval <- apply(cbind(liver.mouse.eQTL.bayesian.tau$lung_
                pvalue, liver.mouse.eQTL.bayesian.tau$liver_pvalue), 1,
                Ncomb)
377         liver.mouse.eQTL.bayesian.tau$metapval <- metapval
378         # MT Method
379         mtresults <- read.table(paste0("MTeQTLs_ASE_3c.txt"), header =
                TRUE)
380         # Select Gene-SNP pair with minimum P value
381         minmtresults <- sapply(liver.mouse.eQTL.bayesian.tau$ensembl_id,
382              function(x) min(mtresults[mtresults$ensembl_id == as.
                    character(x), "marginalP.liver"]))
383         newmtresults <- data.frame(liver.mouse.eQTL.bayesian.tau$ensembl
                _id, minmtresults, liver.mouse.eQTL.bayesian.tau$eqtl)
384         colnames(newmtresults) <- c("ensembl_id", "marginalp", "eqtl")
385         # Merge MT results with the other results
386         newresults <- liver.mouse.eQTL.bayesian.tau[, c("ensembl_id", "
                lung_pvalue", "liver_pvalue", "metapval", "p.below.0", "eqtl
                ")]
387         newresults$marginalp <- newmtresults$marginalp
388         # Merge results of 6 times randomizations
389         combinded.results <- rbind(combinded.results, newresults)
390         orig_auc[p, 1] <- sebsetn
391         orig_auc[p, 2] <- seedlib[k]
392         orig_auc[p, 3] <- auc(newresults$eqtl, newresults$liver_pvalue)
393         bayesian_auc[p, 1] <- sebsetn
394         bayesian_auc[p, 2] <- seedlib[k]
395         bayesian_auc[p, 3] <- auc(newresults$eqtl, newresults$p.below.0)
396         lung_auc[p, 1] <- sebsetn
397         lung_auc[p, 2] <- seedlib[k]
398         lung_auc[p, 3] <- auc(newresults$eqtl, newresults$lung_pvalue)
399         meta_auc[p, 1] <- sebsetn
400         meta_auc[p, 2] <- seedlib[k]
```

```
401         meta_auc[p, 3] <- auc(newresults$eqtl, newresults$metapval)
402         mt_auc[p, 1] <- sebsetn
403         mt_auc[p, 2] <- seedlib[k]
404         mt_auc[p, 3] <- auc(newresults$eqtl, newresults$marginalp)
405         p <- p + 1
406     }
407     # Calcaculate means for roc curve plotting
408     mean.results <- ddply(combinded.results, .(ensembl_id), summarize,
409         lung_pvalue = mean(lung_pvalue), liver_pvalue = mean(liver_
                pvalue),
410         metapval = mean(metapval), p.below.0 = mean(p.below.0),
                marginalp = mean(marginalp))
411     mean.results$eqtl <- newresults$eqtl
412     # Combine subsampling result
413     aorig_auc <- rbind(aorig_auc, orig_auc)
414     abayesian_auc <- rbind(abayesian_auc, bayesian_auc)
415     alung_auc <- rbind(alung_auc, lung_auc)
416     ameta_auc <- rbind(ameta_auc, meta_auc)
417     amt_auc <- rbind(amt_auc, mt_auc)
418     # ROC plotting
419     rocobj1 <- plot.roc(mean.results$eqtl, mean.results$liver_pvalue,
            main = paste0(sebsetn,
420          " strains"), percent = TRUE, col = "black", legacy.axes = TRUE,
                yaxs = "i")
421     rocobj2 <- lines.roc(mean.results$eqtl, mean.results$p.below.0,
            percent = TRUE, col = "red")
422     rocobj3 <- lines.roc(mean.results$eqtl, mean.results$marginalp,
            percent = TRUE, col = "green")
423     # legend(45,30, legend=c('Original liver', 'Bayesian', 'MT'),
424     # col=c('black', 'red', 'green'), lwd=2, cex = 0.85, bty = 'n')
425     legend("bottomright", legend = c("Conventional liver", "TA-eQTL", "
            MT"),
426          col = c("black", "red", "green"), lwd = 2, cex = 2, bty = "n")
```

```
}
dev.off()
aorig_auc <- data.frame(aorig_auc)
abayesian_auc <- data.frame(abayesian_auc)
alung_auc <- data.frame(alung_auc)
ameta_auc <- data.frame(ameta_auc)
amt_auc <- data.frame(amt_auc)
aorig_auc$methods <- "Conventional liver"
abayesian_auc$methods <- "TA-eQTL"
alung_auc$methods <- "Conventional lung"
ameta_auc$methods <- "Meta"
amt_auc$methods <- "MT"
comauc <- rbind(aorig_auc, abayesian_auc, amt_auc, ameta_auc, alung_auc)
write.table(comauc, "comauc0929.txt")
# comauc <- read.table("comauc0929.txt")
comauc$methods <- factor(comauc$methods, levels = unique(as.character(
    comauc$methods)))
## Gives count, mean, standard deviation, standard error of the mean,
## and confidence interval (default 95%).  data: a data frame.
## measurevar: the name of a column that contains the variable to be
## summariezed groupvars: a vector containing names of columns that
## contain grouping variables na.rm: a boolean that indicates whether to
## ignore NA's conf.interval: the percent range of the confidence
## interval (default is 95%)
summarySE <- function(data = NULL, measurevar, groupvars = NULL, na.rm =
     FALSE,
    conf.interval = 0.95, .drop = TRUE) {
    library(plyr)
    # New version of length which can handle NA's: if na.rm==T, don't
        count
    # them
    length2 <- function(x, na.rm = FALSE) {
        if (na.rm)
```

```
457             sum(!is.na(x)) else length(x)
458     }
459     # This does the summary. For each group's data frame, return a
            vector
460     # with N, mean, and sd
461     datac <- ddply(data, groupvars, .drop = .drop, .fun = function(xx,
462         col) {
463         c(N = length2(xx[[col]], na.rm = na.rm), mean = mean(xx[[col]],
464             na.rm = na.rm), sd = sd(xx[[col]], na.rm = na.rm), min = min
                (xx[[col]],
465             na.rm = na.rm), max = max(xx[[col]], na.rm = na.rm))
466     }, measurevar)
467     # Rename the 'mean' column
468     datac <- rename(datac, c(mean = measurevar))
469     datac$se <- datac$sd/sqrt(datac$N)  # Calculate standard error of
            the mean
470     # Confidence interval multiplier for standard error Calculate
471     # t-statistic for confidence interval: e.g., if conf.interval is
            .95,
472     # use .975 (above/below), and use df=N-1
473     ciMult <- qt(conf.interval/2 + 0.5, datac$N - 1)
474     datac$ci <- datac$se * ciMult
475     return(datac)
476 }
477 sumcomauc <- summarySE(comauc, measurevar = "auc", groupvars = c("
        methods", "sebsetn"))
478 # Use sebsetn as a factor rather than numeric
479 sumcomauc2 <- sumcomauc
480 sumcomauc2$sebsetn <- factor(sumcomauc2$sebsetn)
481 aucsubrange <- ggplot(sumcomauc2, aes(x = sebsetn, y = auc, fill =
        methods)) +
482     geom_bar(position = position_dodge(), stat = "identity") + geom_
            errorbar(aes(ymin = min,
```

```
483     ymax = max), width = 0.2, position = position_dodge(0.9)) + xlab("
            Number of strains for analysis") +
484     ylab("AUC") + coord_cartesian(ylim = c(0.5, 1)) + scale_fill_manual(
            values = c("black", "red", "green", "blue", "purple"))
485 comauc$samplingseed <- factor(comauc$samplingseed)
486 sub10 <- subset(comauc, sebsetn == 10)
487 sub15 <- subset(comauc, sebsetn == 15)
488 sub20 <- subset(comauc, sebsetn == 20)
489 sub25 <- subset(comauc, sebsetn == 25)
490 # Mixed model with ramdom effect on samplingseed
491 test10 <- lmer(auc ~ methods + (1 | samplingseed), data = sub10)
492 test15 <- lmer(auc ~ methods + (1 | samplingseed), data = sub15)
493 test20 <- lmer(auc ~ methods + (1 | samplingseed), data = sub20)
494 test25 <- lmer(auc ~ methods + (1 | samplingseed), data = sub25)
495 # Pair comparisons between methods
496 lsmeans(test10, pairwise ~ methods)
497 lsmeans(test15, pairwise ~ methods)
498 lsmeans(test20, pairwise ~ methods)
499 lsmeans(test25, pairwise ~ methods)
500 # Normalized AUC
501 comauc.liver <- subset(comauc, methods == "Conventional liver")
502 comauc.liver$liverauc <- comauc.liver$auc
503 comauc.liver$auc <- NULL
504 comauc.liver$methods <- NULL
505 ncomauc <- merge(comauc, comauc.liver, by = c("sebsetn", "samplingseed")
        )
506 ncomauc$nauc <- ncomauc$auc/ncomauc$liverauc
507 sumncomauc <- summarySE(ncomauc, measurevar = "nauc", groupvars = c("
        methods",
508     "sebsetn"))
509 # Use sebsetn as a factor rather than numeric
510 sumncomauc2 <- sumncomauc
511 sumncomauc2$sebsetn <- factor(sumncomauc2$sebsetn)
```

```
naucsubrange <- ggplot(sumncomauc2, aes(x = sebsetn, y = nauc, fill =
    methods)) +
    geom_bar(position = position_dodge(), stat = "identity") + geom_
        errorbar(aes(ymin = min,
    ymax = max), width = 0.2, position = position_dodge(0.9)) + xlab("
        Number of strains for analysis") +
    ylab(expression("Ratio of AUC vs AUC"["Conventional liver"])) +
        coord_cartesian(ylim = c(0.8,
    1.2)) + scale_fill_manual(values = c("black", "red", "green", "blue"
        , "purple"))
pdf("naucsubrange.pdf", width = 8, height = 4)
print(naucsubrange)
dev.off()
sumncomauc2$min <- NULL
sumncomauc2$max <- NULL
sumcomauc2$min <- NULL
sumcomauc2$max <- NULL
sumcomauc_wide1 <- dcast(sumcomauc2, methods ~ sebsetn, value.var = "auc
    ")
sumcomauc_wide2 <- dcast(sumncomauc2, methods ~ sebsetn, value.var = "
    nauc")
sumcomauc_wide <- cbind(sumcomauc_wide1, sumcomauc_wide2[, 2:5])
sumcomauc_wide <- sumcomauc_wide[, c(1, 2, 6, 3, 7, 4, 8, 5, 9)]
colnames(sumcomauc_wide) <- c("methods", "10_mean", "10_ratio", "15_mean
    ", "15_ratio", "20_mean", "20_ratio", "25_mean", "25_ratio")
write.table(sumcomauc_wide, "sumcomauc_wide0929.txt")
print.xtable(xtable(sumcomauc_wide), type = "latex", file = "combined_
    auc.tex",
    latex.environments = "center", include.rownames = FALSE)
```

B.8 Supplemental codes for Multiple tissue Bayesian analysis

The following code was named as ""2016-09-12mtsubsetanalysis.R" and used in step 6

for subsetting analysis.

```
# delete df.txt if it exists and prepare new analysis of subsampling
if (file.exists("df.txt")) {file.remove("df.txt")}
### Run separately for every tissue tissue = 'Liver'; subset dataset
write.table(sub.mouse.liver.expression.eqtl, file = "expression/liver.expr.txt",
    sep = "\t", row.names = FALSE, quote = FALSE)
# subset liver snp expression data
write.table(sub.BXD.geno.SNP.eqtl.for.liver, file = "genotypes/liver.snps.txt",
    sep = "\t", row.names = FALSE, quote = FALSE)
file.remove("df.txt")
### step 1
tissue <- "liver"
### Running Matrix eQTL ###
library("MatrixEQTL")
### Load genotype info
snps <- SlicedData$new()
snps$LoadFile(paste0("genotypes/", tissue, ".snps.txt"), skipRows = 1,
    skipColumns = 1, sliceSize = 500)
### Load gene expression info
expr <- SlicedData$new()
expr$LoadFile(paste0("expression/", tissue, ".expr.txt"), skipRows = 1,
    skipColumns = 1, sliceSize = 500)
### Load covariates
cvrt <- SlicedData$new()
# cvrt$LoadFile(paste0('covariates/',tissue,'.covariates.txt'),
# skipRows = 1, skipColumns = 1, sliceSize = 500); Load gene locations
geneloc <- read.table(paste0("2016-09-08 ", tissue, ".gene.loc.txt"),
    sep = "\t",
    header = TRUE, stringsAsFactors = FALSE)
### Load SNP locations
snpsloc <- read.table(paste0("2016-09-08 BXD.geno.loc.eqtl.for.", tissue
```

```
    ,
30      ".txt"), sep = "\t", header = TRUE, stringsAsFactors = FALSE)
31 options(MatrixEQTL.dont.preserve.gene.object = TRUE)
32 ### Run Matrix eQTL
33 me <- Matrix_eQTL_main(snps = snps, gene = expr, cvrt = cvrt, output_
       file_name = "",
34      pvOutputThreshold = 0, useModel = modelLINEAR, errorCovariance =
           numeric(),
35      verbose = TRUE, output_file_name.cis = paste0("eQTL_results_AL_",
           tissue,
36          "_cis.txt"), pvOutputThreshold.cis = 1, snpspos = snpsloc,
              genepos = geneloc,
37      cisDist = 1e+06, pvalue.hist = FALSE, noFDRsaveMemory = TRUE)
38 ### Save the number of degrees of freedom for each tissue
39 cat(file = "df.txt", tissue, "\t", me$param$dfFull, "\n", append = TRUE)
40 tissue <- "lung"
41 ### Running Matrix eQTL ###
42 library("MatrixEQTL")
43 ### Load genotype info
44 snps <- SlicedData$new()
45 snps$LoadFile(paste0("genotypes/", tissue, ".snps.txt"), skipRows = 1,
46      skipColumns = 1, sliceSize = 500)
47 ### Load gene expression info
48 expr <- SlicedData$new()
49 expr$LoadFile(paste0("expression/", tissue, ".expr.txt"), skipRows = 1,
50      skipColumns = 1, sliceSize = 500)
51 ### Load covariates
52 cvrt <- SlicedData$new()
53 # cvrt$LoadFile(paste0('covariates/',tissue,'.covariates.txt'),
54 # skipRows = 1, skipColumns = 1, sliceSize = 500); Load gene locations
55 geneloc <- read.table(paste0("2016-09-08 ", tissue, ".gene.loc.txt"),
       sep = "\t",
56      header = TRUE, stringsAsFactors = FALSE)
```

```
### Load SNP locations
snpsloc <- read.table(paste0("2016-09-08 BXD.geno.loc.eqtl.for.", tissue,
    ".txt"), sep = "\t", header = TRUE, stringsAsFactors = FALSE)
options(MatrixEQTL.dont.preserve.gene.object = TRUE)
### Run Matrix eQTL
me <- Matrix_eQTL_main(snps = snps, gene = expr, cvrt = cvrt, output_
    file_name = "",
    pvOutputThreshold = 0, useModel = modelLINEAR, errorCovariance =
        numeric(),
    verbose = TRUE, output_file_name.cis = paste0("eQTL_results_AL_",
        tissue,
        "_cis.txt"), pvOutputThreshold.cis = 1, snpspos = snpsloc,
            genepos = geneloc,
    cisDist = 1e+06, pvalue.hist = FALSE, noFDRsaveMemory = TRUE)
### Save the number of degrees of freedom for each tissue
cat(file = "df.txt", tissue, "\t", me$param$dfFull, "\n", append = TRUE)
### step2 Read df.txt for the list of tissues and degrees of freedom of
### linear models
df <- read.table("df.txt", stringsAsFactors = FALSE)
names(df) <- c("tissue", "df")
show(df)
### List vector for storing Matrix eQTL results
big.list <- vector("list", nrow(df))
### Store gene and SNP names from the first tissue for matching with
### other tissues
genes <- NULL
snps <- NULL
### colClasses for faster reading of Matrix eQTL output
cc.file <- NA
### Loop over tissues
for (t1 in 1:nrow(df)) {
    ### Get tissue name
```

```
85      tissue <- df$tissue[t1]
86      ### Load Matrix eQTL output for the given tissue
87      start.time <- proc.time()[3]
88      tbl <- read.table(paste0("eQTL_results_AL_", tissue, "_cis.txt"),
            header = T,
89          stringsAsFactors = FALSE, colClasses = cc.file)
90      end.time <- proc.time()[3]
91      cat(tissue, "loaded in", end.time - start.time, "sec.", nrow(tbl),
92          "gene-SNP pairs.", "\n")
93      ### set colClasses for faster loading of other results
94      if (any(is.na(cc.file))) {
95          cc.file <- sapply(tbl, class)
96      }
97      ### Set gene and SNP names for matching
98      if (is.null(snps))
99          snps <- unique(tbl$SNP)
100     if (is.null(genes))
101         genes <- unique(tbl$gene)
102     ### Match gene and SNP names from Matrix eQTL output to 'snps' and
103     ### 'genes'
104     gpos <- match(tbl$gene, genes, nomatch = 0L)
105     spos <- match(tbl$SNP, snps, nomatch = 0L)
106     ### Assign each gene-SNP pair a unique id for later matching with
            other tissues
107     id <- gpos + 2 * spos * length(genes)
108     ### Transform t-statistics into correlations
109     r <- tbl$t.stat/sqrt(df$df[t1] + tbl$t.stat^2)
110     ### Record id's and correlations
111     big.list[[t1]] <- list(id = id, r = r)
112     ### A bit of clean up to reduce memory requirements
113     rm(tbl, gpos, spos, r, id, tissue, start.time, end.time)
114     gc()
115 }
```

```
116 rm(t1, cc.file)
117 ### Find the set of gene-SNP pairs present in results for all tissues
118 keep <- rep(TRUE, length(big.list[[1]]$id))
119 for (t1 in 2:nrow(df)) {
120     mch <- match(big.list[[1]]$id, big.list[[t1]]$id, nomatch = 0L)
121     keep[mch == 0] <- FALSE
122     cat(df$tissue[t1], ", overlap size", sum(keep), "\n")
123 }
124 final.ids <- big.list[[1]]$id[keep]
125 rm(keep, mch, t1)
126 ### Create and fill in the matrix of z-scores Z-scores are calculated
127 ### from correlations
128 big.matrix <- matrix(NA_real_, nrow = length(final.ids), ncol = nrow(df)
        )
129 fisher.transform <- function(r) {
130     0.5 * log((1 + r)/(1 - r))
131 }
132 for (t1 in 1:nrow(df)) {
133     mch <- match(final.ids, big.list[[t1]]$id)
134     big.matrix[, t1] <- fisher.transform(big.list[[t1]]$r[mch]) * sqrt(
            df$df[t1] -
135          1)
136     cat(t1, "\n")
137 }
138 stopifnot(!any(is.na(big.matrix)))
139 rm(t1, mch)
140 ### Save the big matrix
141 save(list = "big.matrix", file = "z-score.matrix.Rdata", compress =
        FALSE)
142 ### Save gene names and SNP names for rows of big matrix
143 writeLines(text = genes[final.ids%%(length(genes) * 2)], con = "z-score.
        matrix.genes.txt")
144 writeLines(text = snps[final.ids%/%(length(genes) * 2)], con = "z-score.
```

```
    matrix.snps.txt")
### step3 Set estimation parameters
maxIterations <- 100
### Load big matrix of z-scores
load(file = "z-score.matrix.Rdata")
dim(big.matrix)
### Initialize parameters
{
    param <- list()
    ### K - the number of tissues
    K <- ncol(big.matrix)
    ### Delta - null covariance matrix across tissues
    param$Delta <- matrix(0.05, K, K)
    diag(param$Delta) <- 1
    ### Sigma - signal covariance matrix across tissues
    param$Sigma <- matrix(3, K, K) + diag(K)
    ### P - the vector of probabilities
    param$P <- rep(1/2^K, 2^K)
    ### Psubs - the vector of active eQTLs for each element of P
    Psubs <- vector("list", 2^K)
    for (i in 1:2^K) {
        a <- 2^((K - 1):0)
        b <- 2 * a
        Psubs[[i]] <- as.double(((i - 1)%%b) >= a)
    }
    rm(a, b, i)
    param$Psubs <- Psubs
    rm(Psubs)
    ### loglik - the initial likelihood
    param$loglik <- -Inf
    rm(K)
}
### The function does a single iteration of the estimation procedure
```

```
177 DoIteration <- function(big.matrix, param) {
178     ### extract current model parameters
179     K <- ncol(big.matrix)
180     m <- nrow(big.matrix)
181     Delta <- param$Delta
182     Sigma <- param$Sigma
183     P <- param$P
184     Psubs <- param$Psubs
185     ### The function for matrix power
186     mat.power <- function(mat, pow) {
187         e <- eigen(mat)
188         V <- e$vectors
189         return(V %*% diag(e$values^pow) %*% t(V))
190     }
191     ### Start the timer
192     tic <- proc.time()
193     ### variables to accumulate loglik - likelihood newP - marginal
194     ### probabilities newDelta - the new Delta matrix newSigmaPlusDelta
            - Delta+Sigma
195     cum.loglik <- 0
196     cum.newP <- 0
197     cum.newDelta <- 0
198     cum.newSigmaPlusDelta <- 0
199     ### Do calculations in slices of 10000 gene-SNP pairs
200     step1 <- 100000L
201     for (j in 1:ceiling(m/step1)) {
202         fr <- step1 * (j - 1) + 1
203         to <- min(step1 * j, m)
204         X <- big.matrix[fr:to, , drop = FALSE]
205         ### likelihood for the slice
206         prob <- matrix(0, nrow(X), length(P))
207         for (i in 1:length(Psubs)) {
208             sigma_star <- Delta + Sigma * tcrossprod(Psubs[[i]])
```

```
209             sigma_hfiv <- mat.power(sigma_star, -0.5)
210             sigma_dethfiv <- (det(sigma_star))^(-0.5)
211             w <- (1/(2 * pi)^(K/2)) * (P[i] * sigma_dethfiv)
212             prob[, i] <- exp(log(w) - colSums(tcrossprod(sigma_hfiv/sqrt
                    (2),
213                 X)^2))
214         }
215         cum.loglik <- cum.loglik + sum(log(rowSums(prob)))
216         ### Normalize probabilities for each gene-SNP pair to add up to
                1
217         prob <- prob/rowSums(prob)
218         ### new vector of P - tissue specificity probabilities
219         cum.newP <- cum.newP + colSums(prob)
220         cum.newDelta <- cum.newDelta + crossprod(X * sqrt(prob[, 1]))
221         cum.newSigmaPlusDelta <- cum.newSigmaPlusDelta + crossprod(X *
222             sqrt(prob[, length(P)]))
223     }
224     {
225         ### Calculate Delta from the cumulative sum
226         Delta <- cum.newDelta/cum.newP[1]
227         ### normalize to force the diagonal to 1
228         Delta <- Delta * tcrossprod(sqrt(1/diag(Delta)))
229         ### Same with Sigma
230         Sigma <- cum.newSigmaPlusDelta/tail(cum.newP, 1) - Delta
231         e <- eigen(Sigma)
232         if (any(e$values < 0)) {
233             Sigma <- e$vectors %*% diag(pmax(e$values, 0)) %*% t(e$
                    vectors)
234         }
235     }
236     P <- cum.newP/sum(cum.newP)
237     toc <- proc.time()
238     return(list(Delta = Delta, Sigma = Sigma, P = P, Psubs = Psubs,
```

```
          loglik = cum.loglik,
239         time = toc - tic))
240 }
241 ### The 'paralist' list vector will store model estimates at each
      iteration
242 paralist <- vector("list", maxIterations + 1)
243 paralist[[1]] <- param
244 rm(param)
245 ### Perform up to 'maxIterations' iteration
246 for (i in 2:length(paralist)) {
247     paralist[[i]] <- DoIteration(big.matrix = big.matrix, param =
          paralist[[i -
248         1]])
249     cat(i, "\t", paralist[[i]]$loglik - paralist[[i - 1]]$loglik, "\t",
250         paralist[[i]]$time[3], "\n")
251     if (i > 10)
252         if (paralist[[i]]$loglik < paralist[[i - 1]]$loglik)
253             break
254 }
255 paralist <- paralist[!sapply(paralist, is.null)]
256 ### Save the results
257 save(list = "paralist", file = "paralist.Rdata")
258 ### step4 Parameters
259 local.FDR.threshold <- 1
260 output.file.name <- "MT-eQTLs.txt"
261 ### Load big matrix of z-scores
262 load(file = "z-score.matrix.Rdata")
263 dim(big.matrix)
264 ### Load gene names and SNP names matching the rows of big.matrix
265 gnames <- readLines("z-score.matrix.genes.txt")
266 snames <- readLines("z-score.matrix.snps.txt")
267 ### Load tissue names
268 df <- read.table("df.txt", stringsAsFactors = FALSE)
```

```
269 names(df) <- c("tissue", "df")
270 show(df)
271 ### Load parameter estimates and pick the last one
272 load("paralist.Rdata")
273 param <- tail(paralist, 1)[[1]]
274 ### Number of tissues
275 K <- ncol(big.matrix)
276 m <- nrow(big.matrix)
277 ### The function for matrix power
278 mat.power <- function(mat, pow) {
279     e <- eigen(mat)
280     V <- e$vectors
281     return(V %*% diag(e$values^pow) %*% t(V))
282 }
283 ### Matrix of possible tissue specificity profiles
284 Pmat <- simplify2array(param$Psubs)
285 ### Call eQTLs and save in a file
286 fid <- file(description = output.file.name, open = "wt")
287 writeLines(con = fid, paste0("SNP\tgene\t", paste0("isEQTL.", df$tissue,
288     collapse = "\t"), "\t", paste0("marginalP.", df$tissue, collapse = "
        \t")))
289 ### Do calculations in slices of 10000 gene-SNP pairs
290 step1 <- 10000L
291 cumdump <- 0
292 for (j in 1:ceiling(nrow(big.matrix)/step1)) {
293     fr <- step1 * (j - 1) + 1
294     to <- min(step1 * j, nrow(big.matrix))
295     X <- big.matrix[fr:to, , drop = FALSE]
296     ### likelihood for the slice
297     prob <- matrix(0, nrow(X), length(param$P))
298     for (i in 1:length(param$Psubs)) {
299         sigma_star <- param$Delta + param$Sigma * tcrossprod(param$Psubs
            [[i]])
```

```
300         sigma_hfiv <- mat.power(sigma_star, -0.5)
301         sigma_dethfiv <- (det(sigma_star))^(-0.5)
302         w <- (1/(2 * pi)^(K/2)) * (param$P[i] * sigma_dethfiv)
303         prob[, i] <- exp(log(w) - colSums(tcrossprod(sigma_hfiv/sqrt(2),
304             X)^2))
305     }
306     prob <- prob/rowSums(prob)
307     ### Select tests with eQTLs significant at local.FDR.threshold level
308     keep <- (prob[, 1] <= local.FDR.threshold)
309     if (any(keep)) {
310         marginalProb <- tcrossprod(prob[keep, , drop = FALSE], 1 - Pmat)
311         tissueSpecificity <- t(Pmat)[apply(X = prob[keep, , drop = FALSE
                ],
312             MARGIN = 1, FUN = which.max), ]
313         dump <- data.frame(snames[(fr:to)[keep]], gnames[(fr:to)[keep]],
314             tissueSpecificity, marginalProb, row.names = NULL, check.
                rows = FALSE,
315             check.names = FALSE, stringsAsFactors = FALSE)
316         write.table(dump, file = fid, quote = FALSE, sep = "\t", row.
                names = FALSE,
317             col.names = FALSE)
318     }
319     cumdump <- cumdump + sum(keep)
320     cat("Slice", j, "of", ceiling(nrow(big.matrix)/step1), " eQTLs
            recorded:",
321         cumdump, "\n")
322 }
323 close(fid)
324 ### step5
325 MTeQTLs <- read.table(file = "MT-eQTLs.txt", header = T)
326 liver.ASE.ensembl <- read.table(file = "liver.ASE.ensembl.txt", header =
        T)
327 mouse430aensembl_id <- read.table(file = "2015-12-07 mouse430aensembl_id
```

```
    .txt",
328     header = T)
329 MTeQTLs <- merge(MTeQTLs, mouse430aensembl_id, by.x = "gene", by.y = "
    probe_id")
330 library(data.table)
331 MTeQTLs.min <- data.table(MTeQTLs, key = c("ensembl_id", "marginalP.
    liver"))
332 MTeQTLs.min <- MTeQTLs.min[J(unique(ensembl_id)), mult = "first"]
333 MTeQTLs_ASE <- MTeQTLs.min
334 MTeQTLs_ASE$ASE[MTeQTLs_ASE$ensembl_id %in% liver.ASE.ensembl$ensembl_
    gene_id] <- 1  # 1: ASE; 0: non-ASE
335 MTeQTLs_ASE$ASE[!MTeQTLs_ASE$ensembl_id %in% liver.ASE.ensembl$ensembl_
    gene_id] <- 0  # 1: ASE; 0: non-ASE
336 MTeQTLs_ASE <- subset(MTeQTLs_ASE, select = c("ensembl_id", "marginalP.
    liver",
337     "ASE", "gene", "SNP"))
338 MTeQTLs_ASE_3c <- subset(MTeQTLs_ASE, select = c("ensembl_id", "
    marginalP.liver",
339     "ASE"))
340 write.table(MTeQTLs_ASE_3c, file = "MTeQTLs_ASE_3c.txt", sep = "\t", row
    .names = FALSE,
341     quote = FALSE)
342 # delete df.txt and prepare new analysis of subsampling
343 file.remove("df.txt")
```