FM-eQTL User Guide

Yeji Lee and Xiaoquan Wen

September, 2014

Contents

1 Introduction **4**

1.1 Method Overview 4

1.1.1 Detecting eGenes 5

1.1.2 Fine Mapping of *cis*-eQTLs 5

1.2 Citing FM-eQTL 6

2 Installation **7**

3 Input Files **8**

3.1 Grid File 8

3.2 Data File 9

3.2.1 Input for MVLR Model 9

3.2.2 Input for SSLR Model 9

3.3 Prior Specification 10

4 *cis*-eQTL Mapping with FM-eQTL **11**

4.1 Gene-level Testing 11

4.2 Multiple SNP Fine Mapping Analysis 12

5 Output from FM-eQTL **13**

5.1 Numerical Representation of Configurations 13

5.2 Output from Gene-level Analysis 14

5.3 Output from Multiple SNP Analysis 14

6 *cis*-eQTL Analysis Pipeline **15**

6.1 Pipeline Overview 15

6.2 Data Preparation 15

6.2.1 Input Data Files 16

6.2.2 Define *cis* Region 18

6.2.3 Assemble Data Files 20

6.3 Running Analysis 21

6.3.1 Gene-level Testing 21

6.3.2 Fine Mapping Analysis of *cis*-eQTLs 22

6.4 Utilities 23

6.4.1 Make Grid File 24

1 Introduction

FM-eQTL is designed for multiple eQTL mapping in user-defined *cis* region, using expression and genotype data consisting of one or more subgroups. Subgroups in the eQTL data may represent tissues, cellular conditions, populations in different application contexts. It has been applied in the NIH Genotype-Tissue Expression (GTEx) project to map eQTLs across multiple tissues. Recently, it has also been applied in cross-population meta-analysis of eQTLs across multiple population groups [**?**].

FM-eQTL implements an efficient two-step procedure: first it screens genes that its region harbor at least one *cis*-eQTL (which we call eGenes), second performs multiple SNP analysis (i.e. analyzing multiple SNP simultaneously) for each eGene. We will refer to this procedure as fine mapping of *cis*-eQTL. FM-eQTL provides a set of software tools to help practitioners prepare data, visualize and interpret analysis results, and can run in a distributed/parallel computing environment.

1.1 Method Overview

FM-eQTL implements a statistical framework discussed in [**?**]. Here, we give a brief description of the model and the inference procedure in the context of *cis*-eQTL mapping. We assume that the genotype and expression data are organized into S pre-defined subgroups. Within each subgroup s, we model the associations between the expression levels ($\boldsymbol{y}_s$) of a target gene and the genotypes of its p *cis*-SNPs using a standard multiple linear regression model,

$$\boldsymbol{y}_s = \mu_s \mathbf{1} + \sum_{i=1}^{p} \beta_{s,i} \boldsymbol{g}_{s,i} + \boldsymbol{e}_s, \ \boldsymbol{e}_s \sim \mathrm{N}(0, \sigma_s^2 \boldsymbol{I}),$$

where $\boldsymbol{g}_{s,i}$ represents the genotype of SNP i in subgroup s, $\boldsymbol{e}_s$ represent the residual errors, μ_s and σ_s^2 denote the subgroup specific intercept and residual error variance, and the regression coefficient $\beta_{s,i}$ characterizes the genetic effect of SNP i in subgroup s. Depending on the study design, FM-eQTL allows different types of correlation structure between the residual errors. If the subgroups are formed by non-overlapping unrelated individuals, the $\boldsymbol{e}_s$'s are regarded independent (e.g., in a meta-analysis of eQTLs). If the subgroups contain the same set of samples (e.g., multiple tissues are sampled from a same set of individuals), FM-eQTL estimates the variance-covariance matrix between $\boldsymbol{e}$'s from the data.

We frame the problem of eQTL mapping in multiple subgroups as a variable selection problem. In particular, FM-eQTL identifies non-zero regression coefficients $\beta_{s,i}$'s jointly across all SNPs and subgroups. Following [**?**] and [**?**], for SNP i, we define its binary indicator vector of association status across subgroups as *configuration*, denoted by $\boldsymbol{\gamma}_i := (\mathbf{1}(\beta_{1,i} \neq 0), ..., \mathbf{1}(\beta_{S,i} \neq 0))$. If a SNP shows associations in multiple subgroups, we assume the non-zero effects are correlated and modeled jointly using the Bayesian prior described in [**?**]. Conditional on observed genotype and expression data, FM-eQTL makes *joint* inference with respect to $\mathbf{\Gamma} := (\boldsymbol{\gamma}_1, ..., \boldsymbol{\gamma}_p)$ using an MCMC algorithm described in below.

1.1.1 Detecting eGenes

FM-eQTL implements a gene-level testing procedure using the method described in [**?**], which performs a Bayesian hypothesis testing for

$$\begin{aligned}
&H_0 : \boldsymbol{\gamma}_1 = ... = \boldsymbol{\gamma}_p = \mathbf{0}, \\
&H_1 : \text{some } \boldsymbol{\gamma}_i \neq \mathbf{0}.
\end{aligned}$$

The null hypothesis asserts no genetic association with expression level for any *cis*-SNPs considered in any subgroups. The evidence from the data for or against each hypothesis is summarized by a gene level Bayes factor.

In [**?**], a permutation procedure is applied to control false discovery rate (FDR). FM-eQTL implements a robust and efficient Bayesian FDR control procedure proposed in [**?**], taking advantage of Bayes factors. Because our implementation of the FDR control procedure does not require any permutations, it achieves desired type I error control at only a fraction of the computational cost.

Upon rejecting the null hypothesis, FM-eQTL classifies the corresponding gene as an eGene. It is recommended to apply fine mapping analysis only on identified eGenes.

1.1.2 Fine Mapping of *cis*-eQTLs

For each identified eGene, FM-eQTL performs fine mapping analysis by jointly evaluating the configurations of all *cis*-SNPs across subgroups using a Bayesian inference procedure.

By default, FM-eQTL assumes an independent and identically distributed prior for each $\boldsymbol{\gamma}_i$ and set $\Pr(\boldsymbol{\gamma}_i = \mathbf{0}) = 1 - \frac{1}{p}$, implying that a single eQTL signal is expected *a priori* in the target *cis*-region, i.e.,

$$\mathrm{E}\left[\sum_{i=1}^{p} \mathbf{1}(\boldsymbol{\gamma}_i \neq 0)\right] = 1.$$

This prior specification reflects the fact that vast majority of the *cis*-SNPs are not associated with the expression levels of any target gene. Furthermore, given the target gene is identified as an eGene, this assumption seems, in average, slightly conservative.

The prior probabilities for the remaining values that $\boldsymbol{\gamma}_i$ can take should depend on the application contexts. For meta-analysis of eQTLs, it seems logical to restrict $\boldsymbol{\gamma}_i \in \{\mathbf{0}, \mathbf{1}\}$: the genuine association signals are always consistent in *all* subgroups. Alternatively, to study tissue or condition specific eQTLs, we recommend estimating the distribution of $P(\boldsymbol{\gamma}_i \neq \mathbf{0})$ using the hierarchical model described in [**?**] by pooling information across all genes genome-wide.

Conditional on a given $\boldsymbol{\Gamma}$ value, the likelihood function, $P(\text{ data } | \boldsymbol{\Gamma})$, can be efficiently and accurately approximated up to a normalizing constant using the Bayes factors derived in [**?**]. Following the Bayes rule,

$$P(\boldsymbol{\Gamma} | \text{ data }) \propto \left(\prod_{i=1}^{p} P(\boldsymbol{\gamma}_i) \right) P(\text{ data } | \boldsymbol{\Gamma}),$$

FM-eQTL implements an Metroplois-Hastings algorithm to compute the posterior distribution of $\boldsymbol{\Gamma}$, which fully conveys the information of associated SNPs and their corresponding configurations in all subgroups.

The MCMC algorithm generates a large set of posterior samples from the target distribution $P(\boldsymbol{\Gamma} \,|\, \text{data})$, from which inference results are summarized. FM-eQTL reports the posterior probability for each sampled $\boldsymbol{\Gamma}$ value.

1.2 Citing FM-eQTL

1. Lee, Y. and Wen, X. (2014) FM-eQTL: Software Pipeline for Fine Mapping *cis*-eQTLs in Multiple Subgroups. (submitted)

2. Wen, X. (2014) Bayesian model selection in complex linear systems, as illustrated in genetic association studies. *Biometrics* 70 (1): 73-83

2 Installation

We provide pre-compiled binary executables for some UNIX based operating systems. Alternatively, one can directly download the source code and compile the desired binary executable.

To do so, a GNU C++ compiler,e.g. GNU g++, and GNU Scientific Library (GSL) are required. To download and install GSL on your system, please refer to http://www.gnu.org/gsl/ for details.

3 Input Files

FM-eQTL processes each gene separately. For each gene, the required input files are

1. a grid file describing prior magnitude and heterogeneity of genetic effects across subgroups
2. a data file containing expression levels and genotypes of *cis*-SNPs

The grid file is typically shared across all genes, and the data file needs to be prepared for each gene. The detailed format descriptions for these two files are given in the following subsections.

We also provide software utility that automatically converts multiple gene data files from MatrixEQTL input format, the details are explained in section 6.

3.1 Grid File

The grid file contains prior specifications for various prior models. In all cases, the grid file always contains a two-column data matrix: the first column always represents the heterogeneity parameter (ϕ) and the second column is used to specify the average effect size parameter (ω). Each row of the grid data matrix provides a unique prior model and different rows can be used to describe different prior heterogeneity levels and/or prior average effect sizes. In the grid file, the first column represents ϕ^2 and the second column represents ω^2.

The following sample of the grid file is used by [**?**] to perform tissue-specific eQTL analysis. The grid assumes five levels of overall prior effects ($\sqrt{\omega^2 + \phi^2} = 0.1, 0.2, 0.4, 0.8, 1.6$ values and two degrees of heterogeneity ($\phi^2/\omega^2 = 0, 0.1$), which provides a comprehensive coverage of many possible scenarios. The order of the grid can be arbitrary.

```
0.00000  0.10000
0.00000  0.20000
0.00000  0.40000
0.00000  0.80000
0.00000  1.60000
0.03162  0.09487
0.06325  0.18974
0.12649  0.37947
0.25298  0.75895
0.50596  1.51789
```

3.2 Data File

The data file contains expression levels and genotype information required by FM-eQTL analysis. The formats are slightly different for MVLR and SSLR because of their difference in experiment design. In the former case, the genotypes are shared across all subgroups, whereas in the latter case, the genotypes across subgroups are completely non-overlapping. In both cases, the data files are in text format and contains two consecutive sections for expression levels (phenotype) and genotype, respectively. Missing values in phenotype and genotype data are allowed, and they should be represented by "NA".

3.2.1 Input for MVLR Model

For MVLR model in the phenotype section, each line contains a list of expression levels of all individuals in a subgroup, i.e.,

```
pheno group_id exp_ind_1 exp_ind_2 ... exp_ind_n
```

The leading "pheno" is a keyword that indicates the line encodes expression levels. The group_id field is a character string that uniquely labels a specific group. Note that the group IDs should be consistently used in the data file. The following entries are numerical values of expression levels of individual 1 to n for the target gene in the indicated subgroup.

The genotype section directly follows the phenotype section. Each line contains the genotypes of a SNP, i.e.,

```
geno snp_id  geno_ind_1 geno_ind_2 ... geno_ind_n
```

The leading "geno" is a keyword that indicates the line encodes genotypes. The snp_id field contains a character string that denotes the ID of a SNP. The remaining entries are genotypes of the target SNP for individual 1 to n coded in dosage format (i.e, 0,1 or 2, or any fractional numbers between $[0, 2]$ if the genotype is imputed).

For s subgroups, p SNPs and n individual samples, the data file for MVLR model should contain $(s+p)$ lines each with $(n+2)$ entries.

3.2.2 Input for SSLR Model

For SSLR model in the phenotype section, each line contains a list of expression levels of all individuals in a subgroup, i.e.,

```
  pheno  pheno_id group_id exp_ind_1 exp_ind_2 ... exp_ind_n
```

The leading "pheno" is a keyword that indicates the line encodes expression levels. The pheno_id field contains a character string that denotes the name of the phenotype. (Note this entry is NOT in MVLR model input data) The group_id field is a character string that uniquely labels a specific group. Note that the group IDs should be consistently used in the data file. The following entries are numerical values of expression levels of individual 1 to n for the target gene in the indicated subgroup. Note, because each subgroup can have different number of individuals, the length of each line in this section generally differs.

The genotype section directly follows the phenotype section. Each line contains the genotype information of a SNP for samples in a subgroup, i.e.,

```
  geno snp_id  group_id geno_ind_1 geno_ind_2 ... geno_ind_n
```

The leading "geno" is a keyword that indicates the line encodes genotypes. The snp_id field contains a character string that denotes the ID of a SNP. The additional group_id field indicates the particular subgroup in which genotypes are measured. The remaining entries are genotypes of the target SNP for individual 1 to n in the subgroup coded in dosage format (i.e, 0,1 or 2, or any fractional numbers between $[0, 2]$ if the genotype is imputed).

For s subgroups, p SNPs and n individual samples, the data file for MVLR model should contain $s \times (p + 1)$ lines. The lines are not in equal length: for a line describing a genotype/phenotype for a subgroup with n_s samples, the length should be $(n_s + 3)$.

3.3 Prior Specification

FM-eQTL provides two mechanisms in specifying prior distribution for configuration $\boldsymbol{\gamma}$.

For s subgroups, $\boldsymbol{\gamma}$ can take 2^s different values, and we use integers 0 to $2^s - 1$ to represent these configurations (see section 5.1 for details). The general way to specify a prior is to use a text file, an example of the prior file is shown below

```
pi1  1e-4
config 0.25 0.25 0.50
```

The first line, leading with keyword `pi1` specifies the probability $\Pr(\boldsymbol{\gamma} \neq \mathbf{0})$, equivalently, $\Pr(\boldsymbol{\gamma} = \mathbf{0}) = 1-$ `pi1`. Note, that `pi1` can be left unspecified, and in such case, a default value $1/p$ is assigned where p is the number of *cis*-SNPs for a target gene.

The line starting with keyword "`config`" specifies the probability $\Pr(\boldsymbol{\gamma} = i \mid \boldsymbol{\gamma} \neq \mathbf{0})$ for all non-null configurations $i = 1, ..., 2^s - 1$ sequentially. Note that the numerical values on this line must sum up to 1.

In addition, FM-eQTL recognizes command line option "*-meta*" which represents the prior specification `pi1` $= 1/p$ and $\Pr(\boldsymbol{\gamma} = 2^s - 1 \mid \boldsymbol{\gamma} \neq 0) = 1.0$. This prior is appropriate for meta-analysis where the genuine eQTL is assumed to be consistent across all subgroups. We will call this prior as the "meta-analysis prior" henceforth.

In mapping eQTL across multiple tissues/cellular environments, we recommend that set `pi1` $= 1/p$ and estimate `config` values from software package eQTLBMA.

4 *cis*-eQTL Mapping with FM-eQTL

4.1 Gene-level Testing

The commands to compute the gene-level Bayes factor for a specified gene are as follows: for SSLR

```
    sbams_sslr -d data_file -g grid_file -ga
                     -n gene_name [ -meta | -p prior_file ]
                     -o output
```

for MVLR, the command is almost identical, except that the name of the executable "sbams_sslr" needs to be replaced by "sbams_mvlr". The command line options are

- *-ga*: required for gene-level analysis
- *-d data_file*: required, specification of the data file
- *-g grid_file*: required, specification of the grid file
- *-meta* | *-p prior_file* : required, specification of priors
- *-o output_file*: optional, specification of the output file. By default, the program output goes to screen.

To generate the commands for all interrogated genes, users can initiate the command from `fm_eqtl` as follows

```
     fm_eqtl -ga -p params.ga
```

where parameter file "params.ga" specifies the necessary input options. An example of the parameter file for gene-level analysis is given below:

```
<Contents of params.ga>
FMEQTL_BIN /usr/bin/fmeqtl/bin/
CLUSTER_MODE batch
MODEL SSLR

EQTL_DATA /net/home/eqtl_data/*.dat
GRID_FILE /net/home/grid_file
USE_META  1
```

In this example, the required data files of all the genes are put into directory `/net/home/eqtl_data/`. For each gene, the gene name is coded in the data file name and is represented by the wild card in `EQTL_DATA` option. The output from `fm_eqtl` contains the gene-level analysis commands for all the genes specified, and ready to be executed either in a cluster or a single workstation.

4.2 Multiple SNP Fine Mapping Analysis

The command to run the MCMC algorithm for fine mapping *cis*-eQTLs for a given gene using SSLR model is as follows

```
 sbams_sslr -d data_file -g grid_file -fm
                   [-meta | -p prior_file]
                   -b burnin_step -r repeat_step -o output
```

For MVLR, replace `sbams_sslr` with `sbams_mvlr`. The command line options are

- *-fm*: required for fine mapping analysis
- *-d data_file*: required, specification of the data file
- *-g grid_file*: required, specification of the grid file
- *-meta* | *-p prior_file* : required, specification of priors
- *-b burnin_step*: optional, specification of burnin steps in MCMC. The default setting runs 25,000 burnin steps

- *-r repeat_step*: optional, specification of MCMC repeats. The default setting runs 50,000 repeats
- *-o output_file*: optional, specification of the output file. By default, the program output goes to screen.

Similar to gene-level analysis, users can use `fm_eqtl` to generate the batch command for all the genes that need to be fine mapped. The command syntax is

```
fm_eqtl -fm -p params.fm
```

An example of parameter file "`params.fm` is given below.

```
<Contents of params.fm>
FMEQTL_BIN /usr/bin/fmeqtl/bin/
CLUSTER_MODE batch
MODEL SSLR

EQTL_DATA /net/home/eqtl_data/*.dat
GRID_FILE /net/home/grid_file
USE_META  1
BURNIN    50000
REPEAT   100000
```

5 Output from FM-eQTL

5.1 Numerical Representation of Configurations

In FM-eQTL, we use a decimal number to represent a binary configuration. The conversion between the configuration and the corresponding decimal number follows the rule of converting a number of binary expression to a decimal expression. In particular, the right-most always represents the first subgroup. For 3 subgroups, if an eQTL is only active in the third subgroup, the configuration is represented by a binary string "100" and in turn converted into a decimal number "4". Conversely, a decimal representation "6" indicates a binary string "110" i.e., the eQTL is active in second and third subgroups.

5.2 Output from Gene-level Analysis

The procedure implemented in the gene-level analysis computes a gene-level Bayes factor for each gene. An example output from simulation is given below:

```
Gene    Num_of_cis_SNPs Gene_level_log10_BF Top_SNP Top_SNP_log10_BF
ENSG00000123473   5817   0.561  rs123456  2.719
```

The five columns are: gene name, number of SNPs included in the region, log 10 of gene-level Bayes factor, name of top associated SNP, log 10 of SNP-level Bayes factor for the top associated SNP.

5.3 Output from Multiple SNP Analysis

The output from Multiple SNP Analysis for each gene is organized in a single file, which summarizes results from the MCMC algorithm.

The output is divided into two sections. The first section gives the details of each posterior model sampled during the sampling phase of MCMC runs. For example

```
rank  posterior_prob     posterior_score  log10_BF    model
  1  4.4213e-01          2.017   9.167     [rs45499297:2][rs1554999:1]
```

The consecutive columns indicate the rank (according to the posterior probabilities), the posterior probability, the un-normalized posterior score and the log 10 Bayes factor of the model, whose detailed composition and configuration is given in the last entry. In the above example, the posterior model includes two SNPs (rs45499297 and rs1554999), where rs45499297 is only active in the second subgroup and rs1554999 is only active in the first subgroup. The model has the posterior probability 0.442.

The second section summarizes the marginal posterior inclusion probability (PIP) for each SNP. For example,

```
rank     SNP    config      PIP              Single_SNP_log10_BF
  1 rs45499297    2  9.76737e-01          (1)  0.036   (2)  5.070   (3)  0.981
  2  rs1554999    1  3.68852e-01          (1)  3.755   (2)  0.313   (3)  0.551
```

The columns in each row indicate the rank, SNP name, configuration, PIP and log 10 Bayes factors for each configuration from the single SNP analysis for each SNP.

6 *cis*-eQTL Analysis Pipeline

In addition to the core functionality described in the previous sections, FM-eQTL provides a single software interface to facilitate data preparation, running analysis and a set of utility scripts for post-processing of analysis results.

6.1 Pipeline Overview

FM-eQTL uses a single software interface, the script `fm_eqtl` to help user prepare data and run analysis in a cluster environment. The basic sytax to run `fm_eqtl` is

```
  fm_eqtl -cis_def | -assemble | -prep_eqtlbma | -ga | -fm   -p params
```

More specifically

- *-cis_def*: command option for defining gene-SNP pairs in *cis*
- *-assemble*: command option for assemble required input data files for all the genes to be analyzed
- *-prep_eqtlbma*: command option for generating input files for software package eQTLBMA
- *-ga*: command option for generating job scripts for gene-level testing
- *-fm*: command option for generating job scripts for multiple SNP fine mapping of *cis*-eQTLs
- *-p params*: specification of parameter files

A flow chart of the pipeline is shown in Figure 1.

6.2 Data Preparation

To start data preparation, `fm_eqtl` expects genotype files, expression files , map files for genes and SNPs and a subgroup definition file. For parallel processing, it is highly recommended that genotype files and (gene and SNP) map files are organized by chromosomes. The format of the genotype, expression and map files is the same as what is used in the software package MatrixEQTL, and the format of subgroup definition file differs for SSLR and MVLR models. We give the detailed descriptions in the following sections.

Briefly in the data preparation stage, `fm_eqtl` takes genome-wide data in MatrixEQTL format and convert it to the data file format required by FM-eQTL for each gene.

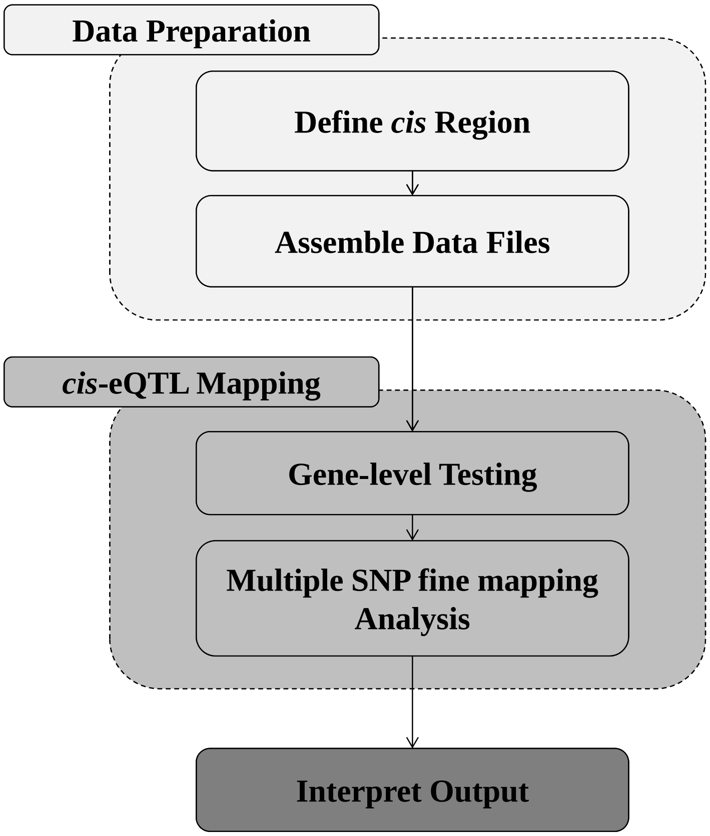

Figure 1: Flow chart of the *cis*-eQTL analysis pipeline implemented in FM-eQTL

6.2.1 Input Data Files

Genotype Information File

The file having genotype information should be separated by white-space or tabs. The first row is assumed to have sample IDs, which are corresponding to IDs in a phenotype information file and a subgroup file. After the first line, each row should contain genotype dosages for each SNP. The first entry of a row denotes SNP's name, which is same with one in maps file. The following entries can take values in numerical or decimal form, while NA indicates a missing value. Missing values will be imputed by the mean of each SNP, but more sophisticated imputation beforehand is strongly recommended. The following is the example of genotype information file :

```
ID id001 id002 id003 id004 id005 ···
snp01 0 0 2 1 NA  ···
snp02 NA 0.5 1.5 2.0 1.0  ···
...
```

FM-eQTL can take either uncompressed or compressed file with gzip (filename extension is ".gz") as input. Also, separated genotype files for each chromosome are recommended.

Phenotype Information File

The phenotype information file is also assumed to be space or tab-delimited, and the sample IDs' are located at the first row. The followings contains expression levels of one gene per each row and the first column consists of genes' name. Expression levels are expected to be adjusted by possible confounders and appropriately transformed to return valid results. Also, files can be separated by each subgroup. Same with genotype information file, missing values can be denoted by NA and the mean imputation will be performed. However, it is still recommended to handle missing data before using FM-eQTL. The example of phenotype information file is as follows :

```
ID id001 id002 id003 id004 id005 ···
gene01 -0.8414491 -1.4945668  0.1981837  NA -1.8755480  ···
gene02 0.9864335 -0.2313034 NA -0.3701193 -2.1197308  ···
...
```

Map Files Map files for genes and SNPs without a header are required, and they should be separated by chromosome. Both map files for genes and for SNPs are required 3 columns : gene name (SNP name), chronosome and basepair position. For each gene(SNP), only one position must be assigned. For position of genes, using the transcription start sites of genes are recommended. Note, MatrixEQTL uses both TSS and TES to anchor a gene, in our case, we allow this format (i.e., multiple anchor points), however, only the last column is used to anchor a gene.

```
<Example of a map file for genes>
gene01 chr1 100000
```

```
gene02 chr1 100050
gene03 chr1 100070
...
<Example of a map file for SNPs>
snp01 chr1 100003
snp02 chr1 100057
snp03 chr1 100081
...
```

Subgroup File: SSLR Model

Subgroup file should have 2 columns and no header. The first column must have sample IDs in genotype information file and phenotype information file, but the order of ID does not need to be the same with them. The second one has subgroup information.

```
id001 group01
id002 group02
id003 group03
...
```

Subgroup File: MVLR Model

In MVLR model, a same set of individuals are presented in all subgroups. We use a different format to define individual and subgroup IDs. More specifically, the file contains two lines, the first line defines subgroup IDs and second line defines individual IDs. For example,

```
group  s1 s2 s3
id id001 id002 id003 ...
```

Each line is lead by the keyword "group" or "id".

6.2.2 Define *cis* Region

The step 1 of converting genome-wide input data into the format required by FM-eQTL is to identify all the gene-SNP pairs for each gene. To do so, FM-eQTL requires a definition of *cis* region for each gene and automatically searches the SNPs that falling into the corresponding *cis* region of each gene.

In FM-eQTL, the *cis* region of a gene is defined by a genomic region centered at the anchor point provided in the gene map file. All the genes has the same region length specified by a user defined radius.

The command to run *cis* region defining option in FM-eQTL is:

```
  fm_eqtl -cis_def  -p params.cis_def
```

In the end, `fm_eqtl` generates a single or multiple batch command script (depending on user specified option), which is ready for submitting to a computing cluster for executions.

The content from an example parameter file "`params.cis_def`" is shown below:

```
<Contents of params.cis_def>
FMEQTL_BIN fmeqtl/bin/
OUTPUT_BATCH  1

CIS_RD 200000
GENE_MAP  chr*.gene.map
SNP_MAP   gene_map/chr*.snp.map
CIS_DEF_DIR  cis_map/
```

The options to be specified in parameter file are:

- `FMEQTL_BIN`: specification of the directory that FM-eQTL binaries are installed
- `OUTPUT_BATCH`: number of batch command files as output. if not specified, the default value is set to 1, the recommended maximum batch number is the number of chromosomes to be processed.
- `CIS_RD` : definition of *cis*-region in the unit of base-pair(bp). Any SNP located within ±value in this parameter will be selected.
- `GENE_MAP` : specification of map file for genes. Wildcard character is allowed when multiple chromosomes are handled simultaneously.
- `SNP_MAP` : specification of map file for SNPs. Same as `GENE_MAP` parameter, wildcard character is allowed.
- `CIS_DEF_DIR` : specification of the directory that saves the output files, which saves *cis*-pairs of SNP and gene. If the directory does not exist, then the program will automatically make one.

We highly recommend that the gene and SNP maps are organized by chromosomes. It would allow parallel processing of all chromosomes simultaneously. The output files are named as "chr*.cis.snp" and saved in the specified `CIS_DEF_DIR`, these files will be used in the assemble step.

6.2.3 Assemble Data Files

Once all gene-SNP pairs are defined in the *cis_def* step, FM-eQTL can assemble the genotype and expression data into required format for analysis.

The following command produces batch scripts that are ready to be submitted to a computing cluster for assembling data files for all genes specified.

```
fmeqtl -p params.assemble -assemble
```

Based on the commonly applied pre-processing steps, we recommend that expression data are divided into subgroups, and genotype data are split by chromosomes. Nonetheless, it is perfectly fine that these data files are further divided (e.g., expression data maybe grouped by chromosome × subgroup combination) or combined. FM-eQTL allows using wildcards in the parameter file. In particular, wildcard "*" represents different chromosomes, and wildcard "#" represents different subgroups.

An example parameter file, `params.ga`, is provided below:

```
<Contents of params.assemble>
FMEQTL_BIN fmeqtl/bin/
OUTPUT_BATCH  100
MODLE SSLR

CIS_DEF_MAP  cis_map/chr*.cis.map
EXPR_DATA    exp_dat/exp.#.dat
GENO_DATA    geno/geno.chr*.gz
SUBGRP_DEF   sub.def.sslr
ASSEMBLE_DIR  eqtl_dat/
```

The options to be specified in the parameter file are:

- `FMEQTL_BIN`: specification of the directory that FM-eQTL binaries are installed

- `OUTPUT_BATCH`: number of batch command files as output. if not specified, the default value is set to 1.
- `MODEL`: the model to be used, either MVLR or SSLR is required.
- `CIS_DEF_MAP` : specification of files that defines *cis*-regions. This file should be generated from the previous step. Wildcard character is allowed to denote multiple chromosomes.
- `EXPR_DATA` : specification of expression phenotype file. In the above example, expression levels are grouped by subgroups. Within each subgroup file, all genes from all chromosomes are included.
- `GENO_DATA` : specification of genotype data file. In the above examples, the genotype data are grouped by chromosomes. In each genotype data, the individuals from all subgroups are included.
- `SUBGRP_DEF` : specification of subgroup file. Note, the format of the files differs for MVLR and SSLR models. The subgroup files can define only a subset of IDs contained in genotype/expression data, the IDs unspecified in the subgroup file will not be included in the analysis.
- `ASSEMBLE_DIR` : specification of the directory that saves the output files. If the directory does not exist, then the program will automatically make one.

Upon successfully running the output command scripts from FM-eQTL, data files for all genes to be analyzed are output into the directory specified by `ASSEMBLE_DIR` with the naming convention "`gene_name.dat`"

6.3 Running Analysis

6.3.1 Gene-level Testing

We highly recommend to perform gene-level testing prior to multiple SNP fine mapping analysis.

The following command produces batch scripts that are ready to be submitted to a computing cluster for computing gene-level Bayes factors for all genes specified.

```
fmeqtl -p params.ga -ga
```

An example parameter file, `params.ga`, for gene-level analysis is provided below:

```
<Contents of params.ga>
FMEQTL_BIN fmeqtl/bin/
```

```
OUTPUT_BATCH  100
MODLE SSLR

EQTL_DATA  eqtl_dat/*.dat
GRID_FILE   grid
PRIOR_SPEC  prior
GA_DIR     ga_rst
```

The options to be specified in the parameter file are:

- FMEQTL_BIN: specification of the directory that FM-eQTL binaries are installed
- OUTPUT_BATCH: number of batch command files as output. If not specified, the default value is set to 1. In general, at this stage, the number should be set according to the number of CPU cores available to the analysis.
- MODEL: the model to be used, either MVLR or SSLR is required.
- EQTL_DATA: specification of assembled data files. Wildcard "*" represents the names of the genes .
- GRID_FILE: specification of grid file having prior values of subgroup heterogeneity and average effect size.
- PRIOR_SPEC: specification of prior file.
- GA_DIR : specification of directory for output files. If the directory does not exist, then the program will automatically make one.

Upon successfully running the output command scripts from FM-eQTL, an output file containing a single line for each gene is generated in the specified GA_DIR.

6.3.2 Fine Mapping Analysis of *cis*-eQTLs

The following command produces batch scripts that are ready to be submitted to a computing cluster for multiple *cis*-eQTL analysis of all genes specified.

```
fmeqtl -p params.fm -fm
```

An example parameter file, params.fm, for gene-level analysis is provided below:

```
<Contents of params.ga>
FMEQTL_BIN fmeqtl/bin/
OUTPUT_BATCH  100
MODLE SSLR

EQTL_DATA  eqtl_dat/*.dat
USE_SUBSET   egene.list
PRIOR_SPEC   prior
GRID_FILE  grid
FM_DIR     fm_rst
```

The options to be specified in the parameter file are:

- `FMEQTL_BIN`: specification of the directory that FM-eQTL binaries are installed
- `OUTPUT_BATCH`: number of batch command files as output. If not specified, the default value is set to 1. In general, at this stage, the number should be set according to the number of CPU cores available to the analysis.
- `MODEL`: the model to be used, either MVLR or SSLR is required.
- `EQTL_DATA` : specification of assembled data files. Wildcard "*" represents the names of the genes .
- `USE_SUBSET`: specification of a subset of genes for fine mapping analysis. If not specified, all genes in the `EQTL_DATA` directory are analyzed. The subset file contains a single column of gene names.
- `PRIOR_SPEC`: specification of prior file.
- `GRID_FILE` : specification of grid file having prior values of subgroup heterogeneity and average effect size.
- `FM_DIR` : specification of directory for output files from fine mapping analysis. If the directory does not exist, then the program will automatically make one.

Upon successfully running the output command scripts from FM-eQTL, output result files for specified genes are generated in the specified `FM_DIR` directory.

6.4 Utilities

FM-eQTL also provides a set of utility scripts, in perl or R, to help users format data, and explore the results.

6.4.1 Make Grid File

The script make_grid help generate the grid file in the analysis by specifying the different levels of overall effect sizes, $\sqrt{\phi^2 + \omega^2}$, and correlation coefficients, $\frac{\omega^2}{\omega^2+\phi^2}$. For example, to generate the grid file shown in section 3.1, we simply make an input file "grid.in" as follows

```
<Contents of grid.in>
size 0.1 0.2 0.4 0.8 1.6
corr 1 0.9
```

By running the script

```
make_grid  grid.in
```

the desired 5×2 grid will be generated.