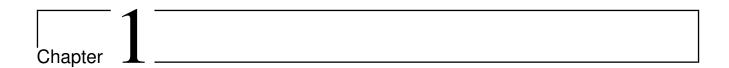
GTEx multi-tissue eQTL analysis

Gao Wang

Last updated: November 16, 2015



General Information

1.1 Release & Analysis V4

Obsolete

- GTEx raw and normalized expression and PEER factors data:
 - ▶ ftp://ftp.broadinstitute.org/GTEx_Analysis_2014-06-13/expressionCovariateSandbox/
 - ▶ There is a README file in this folder that contains description of data
- GTEx summary statistics formatted to eqtlbma input, by Sarah:

Midway /project/mstephens/gtex/preprocessing/june2014/sumstats

• GTEx summary statistics from eqtlbma by Sarah:

Midway /project/mstephens/gtex/analysis/june2014/sumstats

- GTEx genotypes
 - ▶ /project/mstephens/gtex/preprocessing/june2014/omni451.vcf.gz
- GTEx expression level

Midway /project/mstephens/gtex/preprocessing/june2014/GTEx_Analysis_2014-06-13_expression_provisional/

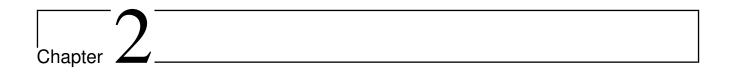
1.2 Release V6

Input and summary statistics from MatrixEQTL. See this page ¹ for details.

 $^{^1} this\ page\ https://github.com/stephenslab/lab-resource/blob/master/data/gtex-v6-eqt1.md$

1.2.1 Why covariates not removed from expression values

"One reason for this is because this is how Matrix eQTL accepts input data. I realize that corrected expression values could be valuable to people and would certainly be willing to provide them, but I do not have a method. Although PEER does output a residuals file, I can only get PEER to run robustly by filtering low expressed genes and opting not to use explicit covariates as input to PEER. The solution would be to use the transformation that Matrix eQTL uses internally. ..." [David DeLuca, Broad institute]



GTEx Summary Statistics

Computations are done on the Midway cluster, using snakemake.

Formatting GTEx Summary Statistics V6

2.1.1 Convert tissue specific results to matrices

For ease of storage / access / numeric operations I convert the summary statistics from plain text file to HDF5 format.

cd src/snakemake/workflows/BMAConfigModel snakemake sumstat_to_h5



I ended up having to perform the entire data conversion on SSD on my desktop computer using. The pytable version of HDF5 implementation takes 2h to convert each tissue but on Midway when HDF5 file size grows to over 1GB the disk I/O is unacceptable. There must be tweaks but not looking into it for the moment. Implementations with h5py for the same purpose is way faster in data conversion (30min each file, performance on Midway is also acceptable) but the resulting data file size is 1.4 times of pytables implementation, even both set to zlib9 compression - thus I stick with pytable. Once converted, it takes almost zero seconds to query data from the HDF5 files.

Merge data across tissues 2.1.2

The files are merged in batches.

snakemake prepare_merge_batch snakemake merge_h5

It takes on average 3 hours per batch to complete. Merged number of genes per batch see log/2015-06-18-merger-log.txt. There are a total of 38933 genes.



I was suspicious to see only 38933 genes in v6. I decide to check against the input data as well as the intermediate HDF5 files I generated

```
compare gene number
snakemake count_genes_from_data
snakemake count_genes_from_h5
```

Counting and comparing results from both sources, I found there are indeed only 38933 genes in the output. So we are good.

Select the "best" and "null" gene-snp pairs 2.1.3

For each gene I select its max and up to 3 null gene-snp pairs, for use of calibrating Sarah's model

snakemake sample_max_null

Creating R Interface for the New Format 2.2

Since the release did not use rsID to name SNPs I create a database to match SNP ID with dbSNP names, to use with R to extract proper SNP-gene by rsID.

```
wget ftp://ftp.ncbi.nih.gov/snp/organisms/human_9606_b144_GRCh37p13/VCF/All_20150605.vcf.gz
```

Build an annotation database matching SNP ID to rs ID as well as the cis-genes involved of up to $\pm 100,000$ bp, in 50 batches.

```
snakemake prepare_snp_lookup_db
snakemake create_snp_lookup_db
```

It takes on average 25 hours per batch to complete. There are 9794339 out of 10297646 (95.1%) SNPs with rsID in latest dbSNP (build 144).

2.2.1 Run queries

Now we have the data *.h5 and meta-data *.sqlite in place. I have made a separate note on how to use this data-set. See SumstatsDB.md in this repo.

Reproducing GTEx V6 Cis-eQTL Analysis 2.3

The idea now is to figure out how the matrixEQTL analysis is done exactly in v6 so that eqtlbma will be done on the same basis, i.e., input files and parameters. For this purpose I make a toy dataset for a randomly chosen tissue Adipose Subcutaneous from GTEx consisting of five files for input to matrixEQTL: genotype SNP.txt, expression GE.txt, a file Covariates.txt of covariates and files geneloc.txt and snpsloc.txt with gene and SNP location information. Below is the data preparation command

snakemake prepare_matrix_eqtl_input

Details see below.

2.3.1 Covariates

There are 40 covariates: "C1,C2,C3,InferredCov1, InferredCov2, ... InferredCov35, gender, Platform". Not all are included in every analysis – inclusion depends on sample size. See the README for details.

2.3.2 SNP data

SNP matrix

Input genotype data are large. Here I just take the first 500 SNPs from the tissue data for verification purpose.



Warning

SNP data released has a lot of imputed data. Cautions are to be taken when comparing with previous eqtlbma analysis (have to make sure the input data are the same).

SNP location

This is required input for matrixEQTL but is not provided. Need to create it from SNP ID's.

Expression data 2.3.3

To focus only on genes paired with the 500 SNPs I get a list of genes via bedtools closest and only extract expression data for these genes. First I create bedtools input and run bedtools to search for genes within 1000bp range to the SNPs of interest. Then I extracted expression data for selected genes.

2.3.4 Gene location data

Location data is extracted for the selected genes.

2.3.5 Find overlapping gene-snp pairs between this toy set and GTEx v6 summary statistics

```
snakemake find_genes_with_sumstats
```

2.3.6 Run matrixEQTL on this toy set

Once the input is prepared it is straightforward to just download the example R script (below, also committed to git repo) and run it, just remember to set the correct data path and change emitted p-value threshold to 1 (pvOutputThreshold=1).

http://www.bios.unc.edu/research/genomic_software/Matrix_eQTL/R.html

To obtain value for one gene-snp pair:

and to verify value for gene-snp pairs:

I verified a few other gene-SNP pairs, all agree. So I can claim I've figured out the input data and parameters used for the GTEx analysis!

2.4 Update eqtlbma Analysis for Some Gene-snp Pairs and Verify with GTEx matrixEQTL

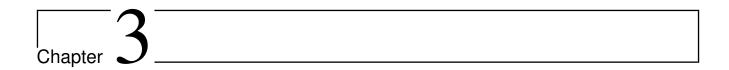
Here is the command for generating summary statistics via eqtlbma_bf:

snakemake eqtlbma_toy

```
gene
                         snp
                                                  betahat.geno
                                                                                                      betapval.geno
ENSG00000177757.1
                                                              1_662622_G_A_b37
                                                                                                                                                          -2.588170e-04
                                                                                                                                                                                                           9.985439e-01
ENSG00000177757.1
                                                                         1_676127_C_T_b37
                                                                                                                                                         7.151412e-02 7.744839e-01
ENSG00000177757.1
                                                                       1_691541_AT_A_b37
                                                                                                                                                         -9.077870e-02 4.383949e-01
ENSG00000177757.1
                                                                       1_693625_T_C_b37
                                                                                                                                                         -5.757815e-02 7.415218e-01
ENSG00000177757.1
                                                                      1_693731_A_G_b37
                                                                                                                                                         -1.080375e-02 9.255429e-01
                                                                                                                                                         5.502301e-01 2.327584e-02
                                                                      1_697411_G_GA_b37
ENSG00000177757.1
ENSG00000177757.1
                                                                      1_701835_T_C_b37
                                                                                                                                                         7.243649e-01 3.168885e-03
ENSG00000177757.1
                                                                      1_704367_T_C_b37
                                                                                                                                                        1.898470e-01 3.756234e-01
ENSG00000177757.1
                                                                       1_705882_G_A_b37
                                                                                                                                                         -1.595642e-01 3.141656e-01
Check with MatrixEQTL
gg = 'ENSG00000177757.1'
for (ss in c('1_662622_G_A_b37', '1_676127_C_T_b37', '1_691541_AT_A_b37', '1_693625_T_C_b37', '1_693731_A_G_b37')) print(me%all%eqtl
s[\label{lem:special} s[\label{lem:special} s[\label{lem:special}] s[\label{lem:special} s[\label{lem:special}] s[\label{lem:special}] s[\label{lem:special}] s[\label{lem:special} s[\label{lem:special}] s[\label{lem:special}] s[\label{lem:special}] s[\label{lem:special} s[\label{lem:special}] s[
                                                                                                                                                                              - OUTPUT
```

```
pvalue
                 gene
                                 snps
                                              bet.a
5493 ENSG00000177757.1 1_662622_G_A_b37 -0.000258817 0.9985439
                 gene
                                snps
                                            beta
                                                    pvalue
4593 ENSG00000177757.1 1_676127_C_T_b37 0.07151412 0.7744839
                 gene
                                 snps
                                             beta
                                                     pvalue
3047 ENSG00000177757.1 1_691541_AT_A_b37 -0.0907787 0.4383949
                 gene
                                snps
                                             beta
                                                     pvalue
4454 ENSG00000177757.1 1_693625_T_C_b37 -0.05757815 0.7415218
                 gene
                                 snps
                                            beta
                                                     pvalue
5200 ENSG00000177757.1 1_693731_A_G_b37 -0.01080375 0.9255429
```

Thus eqtlbma_bf agrees with MatrixEQTL when input agree. Good! We'll have to update the analysis using new input data.



GTEx Summary Statistics Database

3.1 Databases

3.1.1 Multi-tissue summary stats

Summary statistics from the GTEx v6 analysis are stored in MatrixEQTLSumStats.h5 (130GB, in HDF5 format). It contains 38,933 genes (less than previous version! See the list /project/mstephens/datal each gene has several summary statistics matrix of dimension $N_{cisSNPs} \times N_{tissues}$. For this release $N_{tissues} = 44$. For MatrixEQTL the summary statistics are $\hat{\beta}$, T-stat and P-value.

```
MatrixEQTLSumStats.h5

/ENSG00000008735.10 'MatrixEQTL summary statistics of GTEx Release 2015.04.15'
/ENSG00000008735.10/beta (5636, 44)
/ENSG00000008735.10/p-value (5636, 44)
/ENSG00000008735.10/t-stat (5636, 44)
/ENSG00000008735.10/rownames (5636,)
/ENSG00000008735.10/colnames (44,)
...
```

There are two special tables in MatrixEQTLSumStats.h5. One called max, which is the data for the "best" gene-snp pair identified per gene as the snp pair having maximum abs(t-stat) across all snps and all tissues and there is no missing data for any tissue. It contains 16,069 gene-snp pairs. The other called null, which organize gene-snp pair data in the same format as max only that the gene-snp pairs in this table are NOT the best gene-snp pair, but just random samples from all non-best gene-snp pairs. For portability these two tables are also stored separately in MatrixEQTLSumStats.Portable.h5 (58MB, in HDF5 format).

3.1.2 Meta-database

Unfortunately GTEx names cisSNP by their genomic coordinate in Human Genome Assembly hg19 (b37), not by rsID. To make it possible to search with rsID I created another database snp-gene.db (1.6GB, in SQLite format).

```
snp-gene.db
3_52771872_C_T_b37
                                        ENSG00000016864.12, ENSG00000055955.11, ENSG00000055957.6 ...
3_52774530_C_T_b37
                                        ENSG00000016864.12,ENSG00000055955.11,ENSG00000055957.6 ...
                        rs2159644
3_52774835_AT_A_b37
                        rs35270761.rs397785457 ENSG00000016864.12.ENSG00000055955.11.ENSG00000055957.6 ...
```

The 3 columns are

- SNP name in GTEx convention
- rsID
- cis-genes, i.e., genes whose TSS is within 100,000bp up/down-stream of the SNP.

This database will be used prior to searching the summary statistics.



Note

- There may be multiple rsID associated with the same coordinate. Indeed such records exists in dbSNP 144, and eventually these rsIDs should merge into one name (as always have happened in history).
- There are 10,297,646 SNPs in total in this dataset. 9,794,339 of them (95.1%) has rsID in dbSNP 144.

Queries 3.2

I provide a simple R script SumstatQuery. R to make queries on this dataset. It has:

- A function that lists for given rsID the GTEx SNP ID and all its cis-genes.
- A function that loads sumstats data given gene name and GTEx SNP ID.

To use the script, RSQLite and rhdf5 should be installed, e.g.,

```
R
install.packages("RSQLite")
source("http://bioconductor.org/biocLite.R")
biocLite("rhdf5")
```

3.2.1 Example query

The demo script below shows an example to extract information for gene ENSG00000171960.6 and SNP rs6600419, and examples to extract the best/null gene-snp pairs.

```
R
source("/project/mstephens/gtex/scripts/SumstatQuery.R")
# Load data for given gene
dat <- GetSS("ENSG00000171960.6", "/project/mstephens/gtex/analysis/april2015/query/MatrixEQTLSumStats.h5")
print(names(dat))
## [1] "beta"
                               "p-value" "t-stat" "z-score"
dim(dat$"beta")
## [1] 6344 44
dat$"beta"[1:4,1:4]
                                        Adipose_Subcutaneous Adipose_Visceral_Omentum Adrenal_Gland Artery_Aorta
## 1_42124161_C_T_b37
                                                       0.04608095
                                                                                                           NaN NaN 0.06785597
## 1_42124246_A_AC_b37
                                                        -0.03087348
                                                                                                    -0.03222961 -0.1286249 -0.13490575
                                                                                                                              NaN -0.21858459
## 1_42124311_G_A_b37
                                                         0.12400229
                                                                                                      0.08011222
                                                       0.04630340
## 1_42124614_C_T_b37
                                                                                                                                           NaN 0.06785597
                                                                                                                NaN
# Output information for the SNP of interest
dat$"p-value"["1_43124701_A_G_b37", ]
                                  Adipose\_Subcutaneous
                                                                                                Adipose_Visceral_Omentum
 ##
                                                 1.632499e-01
                                                                                                                    3.462833e-02
  ##
                                                Adrenal\_Gland
                                                                                                                      Artery_Aorta
  ##
                                                  8.168959e-01
                                                                                                                      5.600338e-01
                                                                                                                   Artery_Tibial
  ##
                                             Artery Coronary
  ##
                                                 7.659/26e-01
                                                                                                                      7.397466e-01
  ## Brain_Anterior_cingulate_cortex_BA24
                                                                                          Brain\_Caudate\_basal\_ganglia
  ## ...
dat$"z-score"["1_43124701_A_G_b37", ]
                                                                                                Adipose_Visceral_Omentum
                                Adipose\_Subcutaneous
                                                                                                                       -2.11267803
                                                   -1.39422418
                                                Adrenal\_Gland
  ##
                                                                                                                      Artery_Aorta
  ##
                                                   -0.23153929
                                                                                                                         0.58279135
                                             Artery_Coronary
                                                                                                                    Artery\_Tibial
                                                    0.29768626
  ##
                                                                                                                        0.33218889
  ## Brain_Anterior_cingulate_cortex_BA24
                                                                                       Brain_Caudate_basal_ganglia
###
# Load the best gene-snp data
mdat <- GetSS("max", "/project/mstephens/gtex/analysis/april2015/query/MatrixEQTLSumStats.h5")</pre>
dim(mdat$"t-stat")
mdat$"p-value"[1:4,1:4]
mdat$"t-stat"["ENSG00000000419.8_20_49461813_G_C_b37",]
## Is this gene-snp pair most significant in spleen?
dat <- GetSS("ENSG00000000419.8", "/project/mstephens/gtex/analysis/april2015/query/MatrixEQTLSumStats.h5")
idx.to.show <- matxMax(abs(dat$"t-stat"))</pre>
rownames(dat$"t-stat")[idx.to.show[1]]
colnames(dat$"t-stat")[idx.to.show[2]]
###
# Load the "null" gene-snp data
ndat <- GetSS("null", "/project/mstephens/gtex/analysis/april2015/query/MatrixEQTLSumStats.h5")</pre>
dim(ndat$"t-stat")
# Look up GTEx SNP ID
ShowSNP("rs6600419", "/project/mstephens/gtex/analysis/april2015/query/snp-gene.db")
## GTEx SNP ID: 1_43124701_A_G_b37
\textit{\#\# cisGenes: ENSG0000065978.13, ENSG00000164007.6, ENSG00000171960.6, ENSG000002364.1, ENSG00000234917.1, ENSG00000236180.2}
# Look up rs ID given GTEx SNP ID
Show SNP ("1\_43124701\_A\_G\_b37", "/project/mstephens/gtex/analysis/april2015/query/snp-gene.db") = (12.43124701\_A\_G\_b37", "/project/mstephens/gtex/analysis/april2015/query/snp-gene.db") = (12.43124701\_A_G\_b37", "/project/mstephens/gtex/april2015/query/snp-gene/gtex/april2015/query/snp-gene/gtex/april2015/query/snp-gene/gtex/april2015/query/snp-gene/gte
## rsID(s): rs6600419
\textit{\#\# cisGenes: ENSG00000065978.13, ENSG00000164007.6, ENSG00000171960.6, ENSG00000230254.1, ENSG00000234917.1, ENSG00000236180.2}
##
# Create matched training/testing sets
##
```

```
N1 <- 8000
N2 <- 16069
strong.train <- SubsetMatLists(mdat, seq(1, N1))</pre>
strong.test <- SubsetMatLists(mdat, seq(N1 + 1, N2))</pre>
strong.train.genes <- as.character(lapply(strsplit(rownames(strong.train$beta), "_"), function(x) x[1]))
strong.test.genes <- as.character(lapply(strsplit(rownames(strong.test$beta), "_"), function(x) x[1]))
null.genes <- as.character(lapply(strsplit(rownames(ndat$beta), "_"), function(x) x[1]))</pre>
null.train <- SubsetMatLists(ndat, which(null.genes %in% strong.train.genes))</pre>
null.test <- SubsetMatLists(ndat, which(null.genes %in% strong.test.genes))</pre>
```

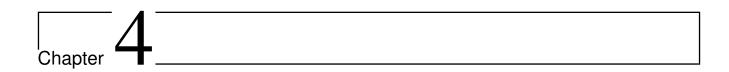
For MatrixEQTLSumStats.h5, for given rsID you first search the GTEx SNP ID via ShowSNP, then load the cisGene of interest via GetSS and extract information using the GTEx SNP ID as the row name key.



If R complains the given row name does not exist, try to flip the DNA strand in the GTEx SNP ID and search again. For example, instead of searching for 1_43124701_A_G_b37 you search for 1_43124701_T_C_b37.

The best gene-snp data can be loaded to memory, via GetSS('max', 'MatrixEQTLSumStats.h5'), or the portable version GetSS('max', 'MatrixEQTLSumStats.Portable.h5')

Same for the "null" gene-snp data, via GetSS('null', 'MatrixEQTLSumStats.h5'), or the portable version GetSS('null', 'MatrixEQTLSumStats.Portable.h5')



Computational Steps for Calculating Summary Statistics Under Quasi-Poisson Model

4.1 First Attempt

Here is the command which breaks the data into 2,000 batches and perform analysis under quasi-Poisson model.

snakemake eqtlbma_batch_poisson

4.1.1 Result

Computation is heavy and not completed (see next section). Here taking a sample batch out of the 2000 batches above, the following is summarized from eqtlbma log file

- Input batch has 28 genes and 113 SNPs
- Genotype and expression of 44 tissues are loaded, vary in sample size, from >70 to 450.
- Tissue specific genes "to keep" vary, but are around 26, 27, 28.
- Number of SNPs (from omni451.vcf.gz) drop greatly after discarding missing values and filter by MAF < 0.05. There are only 10 out of 113 SNPs left.
- Test for each gene-snp pair are done for every batch. There are only a handful such pairs for 2000 batches.

When 5000 batches are created in another test run, some batch may end up fail after loading the data because there will be no qualified SNP left for that batch, from our sample data.

4.1.2 A note on computational time

There will be a 20min "overhead" for each job (loading / extracting data) regardless of per batch size, i.e., we will "waste" 0.3N CPU hours for running in N batches. It is thus best to find a batch neither too small nor to large (if we do not optimize the way data is loaded).

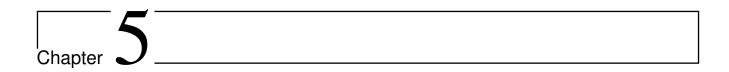
In my test job when only 10 SNPs and <28 genes are analyzed in effect, it takes only 1.38 sec to complete for normal model, evaluating for each tissue about 200 gene-SNP pairs.

However for the same job under quasi-Poission it takes **significantly** computation time. For example 5 hours (10:30 am to 3:30 pm) have passed for the 10 SNPs and 28 genes. Still the progress bar is at 3.75%. I have tried --thread option of eqtlbma_bf and reserve corresponding number of nodes per job on Midway. Somehow the computational node still only use one thread for a job. May have to figure it out but I doubt it's going to help now that we can easily create arbitrary batches.

Here is the eqtlbma_bf command in action for this one batch. For this batch genes_2000_85-_112.bed.gz has 28 lines and snps_2000_85_112.txt.gz has 113 lines.

4.2 What Next?

After discussion with Matthew and William on June 22, 2015 we decide to pause this analysis (from experience of Pritchard's lab the gain from using GLM does not worth it's computational cost).



Analyzing GTEx Using eqtlbma

Pre-processing 5.1

5.1.1 Extract input data from archive

snakemake extract_data

Generate Gene TSS / SNP coordinates files

The coordinate file for genes needs to be prepared. Also unfortunately the release does not come with a list of SNPs (union) involved. Getting such a list is quite a heavy duty. Takes 10min to extract ID's in parallel and 50min to concatenate them into a unique list in bed format.

snakemake prepare_coords



In Sarah's analysis based on an earlier version of data, there are 55,993 genes and 6,856,776 SNPs provided in gene/snp lists. In the v6 release there are 56,318 genes and 10,297,646 SNPs listed; although the sumstats of v6 only has \sim 39,000 genes.

5.1.3 Generate input file lists

snakemake prepare_input_lists

5.2 The Configuration Model

5.2.1 Run eqtlbma_bf analysis in batches

(default: None)

--dry-run

--clean

-e EQTLBMA_PATH Path to an eqtlbma_* executable. (default: None)

analysis. (default: False)

performing analysis. (default: False)

analysis_admin eqtlbma_batch generates batches on the fly and perform eqtlbma_bf analysis in (embarrassing) parallel fashion on batches of genes. To use it,

```
python analysis_admin.py eqtlbma_batch -h
                                                    eqtlbma_batch
usage: analysis_admin.py eqtlbma_batch [-h] -g GENE_COORDS -s SNP_COORDS
                                      [-w N] [-n N_BATCHES] [-b BATCH_ID] -a
                                      ARGS_FILE [--seed N] -e EQTLBMA_PATH
                                      [--dry-run] [--clean]
optional arguments:
 -h, --help show this help message and exit
 -g GENE_COORDS Gene (TSS) coordinate file, in bed.gz format. (default:
 -s SNP_COORDS SNP coordinate file, in bed.gz format. (default: None)
                 Window size. (default: 100000)
 -n N_BATCHES Total number of batches. (default: 1000)
 -b BATCH_ID
                Execute the i-th batch. Program will quit if invalid ID is
                 provided. (default: 1)
 -a ARGS_FILE Path to file containing additional eqtlbma command
                  arguments. (default: None)
                  If specified, a random number will be generated using (N +
 --seed N
                  batch_id) as seed, and the eqtlbma command will be appended
```

Here is the command which breaks the data into 100 batches and perform analysis

a "--seed" argument with the number generated here.

Only generate and save the batch data & commands without

Remove batch gene / snp coords file upon finishing the

```
100 batches analysis
nBatches=100
Model=normal
for i in 'seq $nBatches'; do
echo '#!/bin/bash
    source $HOME/GIT/gtex-eqtls/src/cfg/GTEx.bashrc
    python $SrcDir/python/analysis_admin.py eqtlbma_batch \
     -g $InputDir/tss_coords.bed.gz \
     -s $InputDir/snp_coords.bed.gz \
     -n '"$nBatches"' -b '"$i"' -w 100000 \
     --seed 10086 -e ~/software/bin/eqtlbma_bf \
     -a $ConfDir/eqtlbma.'"$Model"'.txt' |\
sbatch -J eqtlbma_"$Model"_"$nBatches"_"$i" -e $LogDir/eqtlbma_"$Model"_"$nBatches"_"$i".e%j \
       -o $LogDir/eqtlbma_"$Model"_"$nBatches"_"$i".0%j --mem-per-cpu=10000 --time=36:00:00
sleep 1
done
```

See my lab notebook for various failures and fixes running this command. Read on for procedure that succeeded.

5.2.2 Configuration model without permutation

Job submission

```
No permutation for configuration model
nBatches=100
Model=normal.perm0
for i in `seq $nBatches`; do
echo '#!/bin/bash
    source $HOME/GIT/gtex-eqtls/src/cfg/GTEx.bashrc
    python $SrcDir/python/analysis_admin.py eqtlbma_batch \
     -g $InputDir/tss_coords.bed.gz \
     -s $InputDir/snp_coords.bed.gz \
     -n '"$nBatches"' -b '"$i"' -w 100000 \
     --seed 10086 -e ~/software/bin/eqtlbma_bf \
     -a $ConfDir/eqtlbma.'"$Model"'.txt' |\
sbatch -J eqtlbma_"$Model"_"$nBatches"_"$i" -e $LogDir/eqtlbma_"$Model"_"$nBatches"_"$i".e%j \
       -o $LogDir/eqtlbma_"$Model"_"$nBatches"_"$i".o%j --mem-per-cpu=10000 --time=36:00:00
sleep 1
done
```

Run completed under 24 hours, yielding to 45GB output file in various batches. For storage purpose it is best idea to first convert these files in different batches to HDF5 and merge them to one file & archive. Skipping this step for now as I want to move on to the HM and BMA steps. These output are archived as is, though:

```
tar -cvf eqtlbma_bf.normal.perm0.tar *
```

5.2.3 List of summary statistics

The following script produces sumstats lists for all results previously computed:

```
make sumstats lists

SumstatsDir=$ArchiveDir/eqtlbma_bf/July2015

for i in `find $SumstatsDir -maxdepth 1 -name "eqtlbma_bf_normal_*" -type d`; do
    for j in `ls $i/*_sumstats_*.txt.gz`; do
        echo `echo $j | sed 's/\(.*sumstats_\)\(.*\).txt.gz/\(2/g'` $j >> $i.ss.list)

done

done
```

Future analysis can be based on summary statistics instead.

5.3 The Type Model

The raw BF data are fed to the type-model version of the hierarchical model EM algorithm to estimate hyperparameters. Data is 28GB compressed text which may exceed 256GB when all are loaded into RAM (which is what eqtlbma_hm does!).

See my labnotes of July, 2015 to August, 2015 for failures and fixes.