hapQTL User Manual

Yongtao Guan
Baylor College of Medicine
ytguan@gmail.com

Version 0.99
7 July 2014
Revised on 25 June 2015

Contents

1 Copyright

2 What `hapQTL` does

The software, Haplotype Qutatative Loci (hapQTL), is developed and maintained by Yongtao Guan at Baylor College of Medicine, with help from Hanli Xu. Please refer to the paper for details of the method (`http://www.ncbi.nlm.nih.gov/pubmed/24812308`). The `hapQTL` first quantifies genetic relatedness between individuals at an arbitrary marker using local haplotype sharing, and a random effect model is employed to link the genetic relatedness with phenotypes to perform association test at each core marker. Although the test was done at each marker, it is still a haplotype association method because the genetic relatedness at the core marker is quantified through (unspecified) local haplotypes. Comparing to existing haplotype association methods, hapQTL has the following advantages.

- It can directly work with diploid data, no phasing required.
- It doesn't require arbitrary window to define haplotypes, instead, the extend of haplotypes is learned from a statistical model for linkage disequilibrium.
- Each SNP is a core marker for its local haplotypes and its associated test statistics are computed (both Bayes factor and P-value). Thus, our haplotype analysis has the same number tests as the single-SNP analysis.

3 Example

Untar the downloaded file, and one finds two executables: hapQTL-mac (or hapQTL in an early release) for Mac and hapQTL-lin for Linux. There are two subfolders src/ and example/ that contain source files and example files respectively. In the example/ subfolder, there are six example files. geno.txt, geno-raw.txt, pos.txt, pheno.txt, cov.txt, and filter.txt. As their names stated, these are genotypes, raw genotypes, SNP positions, phenotypes, covariates that need to be controlled for, and indicators to remove individuals, respectively. The genotype, position and phenotype files are mandatory, and the cov.txt and filter.txt are optional.

One may perform a testing run using the following command line:

```
./hapQTL -g example/geno.txt -p 1 -pos example/pos.txt \
                 -FILE example/pheno.txt -o test
```

This will generate three files with prefix “test” in a newly created subfolder output/. These files are test.log.txt, test.snpinfo.txt, and test.bf.txt. We will explain these in detail later.

4 Input File Formats

4.1 Genotype file format

Genotypes should be for bi-allelic SNPs, all on the same chromosome. The first two lines should each contain a single number. The number on the first line indicates the number of individuals; the number in the second line indicates the number of SNPs. Optionally, the third row can contain individual identifiers for each individual whose genotypes are included: this line should begin with the string IND, with subsequent strings indicating the identifier for each individual in turn. Subsequent rows contain the genotype data for each SNP, with one row per SNP. In each row the first column gives the SNPs “name” (which can be any string, but might typically be an rs number), and subsequent columns give the genotypes for each individual in turn. Genotypes must be coded in ACGT while missing genotypes can be indicated by NN, ??, or 00 (zero). An example genotype file with 5 individuals and 4 SNPs is below.

```
5
4
IND, id1, id2, id3, id4, id5
rs1, AT, TT, ??, AT, AA
rs2, GG, CC, GG, CC, CG
rs3, CC, ??, ??, CG, GG
rs4, AC, CC, AA, AC, AA
```

Note that plink can convert genotype files from plink format into bimbam format. The option is `--recode-bimbam`.

4.2 SNP position file format

The file contains three columns: the first column is the SNP ID, the second column is its physical location, and the third column contains its chromosome number (optional). It is okay if the rows are unordered, `hapQTL` will sort the SNPs based on their position and chromosome number. `hapQTL` can take multiple position files as input, and duplicate entries are acceptable. If the genotype files contain SNPs across different chromosome, `hapQTL` will sort SNPs based on its chromosome and position. However, we recommend users to run `hapQTL` chromosome by chromosome. The following describes an example file, where the delimit “,” can be changed to “ ”.

```
rs1, 1200,  1
rs2, 4000,  1
rs3, 3320,  1
```

One may insist on running multiple chromosomes together, then the best practice is to stitch the position file together, and add $100,000,000$ unto the SNP positions with each chromosome swtich. For example, three lines in position file [rs11, 1200, 1], [rs22, 2323, 2], and [rs33, 3400, 4] may become [rs11, 1200, 99], [rs22, 100,002323, 99], and [rs33, 200,003400, 99]. The $100,000,000$ increment is because `hapQTL` used prior such that 100,000,000 corresponds to 1 Morgan.

4.3 Phenotype file

Each individual's phenotype value occupies a row. The individuals should have the same order as those in the genotype file. No missing data are allowed. For case control design, use 1 for cases and 0 for controls; we treat the binary phenotypes as quantitative ones in association testing. An example file that contain 5 phenotypes is below:

```
-0.1906689
 0.6430579
 1.0646248
 0.9002399
-0.3561080
```

4.4 Covariates file

Same as phenotype file, each individual occupies a row in the covariates file. The difference is that there are usually multiple columns in this file. The individuals should have the same order as those in the genotype file. No missing data are allowed. If there are missing values for some individuals, we recommend to replace the missing value with the mean of that covariate. An example file that contain three covariates for 5 individuals is shown below:

```
1, -0.6493242,  1.4506556
0,  0.4436516,  0.6158216
0, -1.6376914,  0.7719995
1, -1.4223895, -0.5368858
1, -1.2909029,  1.9088563
```

where the first column may represent sex, and the last two columns may represent two principal components.

4.5 Individual filter file

This file consists of 1 and 0, each occupies one row, and the value is assigned to the individual of the same order in the genotype file. If the filter file were used (see an example below), those individuals are assigned 0 will be removed. Note, this filtering only affects genotype file, not phenotype nor covariate file. Thus it is user's responsibility to remove the corresponding rows in phenotype and covariates files. This appears to be inconvenient at first glance, however, the considerations are following. During the QC procedure, individuals are removed according to genotypes alone, and covariates, which often contain principal components of the remaining individuals, often need to be recomputed accordingly. In addition, the phenotypes of the remaining individuals are usually renormalized after QC.

5 Running hapQTL

First some general comments:

- hapQTL is a command line based program. The command should be typed in a terminal window, in the directory in which hapQTL executable resides.
- The command line should be all in one line: the line-break (denoted by back-slash) in the example is only because the line is too long to fit on one page.
- Unless otherwise stated, the "options" (-g -p -pos -o, etc.) are all case-sensitive.

Now we illustrate how to use hapQTL through examples.

1. **A minimal example**

   ```
   ./hapqtl -g example/geno.txt -p 1 -pos example/pos.txt -FILE example/pheno.txt \
   -DOC example/cov.txt  -C 2 -c 8 -o pref -e 2 -w 40
   ```

   The command line will run EM 2 times with 40 steps each, and use 2 upper-layer clusters and 8 lower-layer clusters. The output filenames begin with "pref."

2. **A more complicated example**

   ```
   ./hapqtl -g example/geno-raw.txt -p example/filter.txt -pos example/pos.txt \
   -w 50 -C 3 -c 10 -mg 200 -exclude-maf 0.01 --exclude-nopos -sem 1 \
   -o pref  -FILE example/pheno.txt -DOC example/cov.txt
   ```

   This command line takes genotype data, filters out individuals according to filter.txt, runs EM once (default) with 50 steps, and uses 3 upper-clusters and 10 lower clusters. The prior LD length is 0.5 centi-Morgan (1/200 Morgan). SNPs whose minor allele frequencies are less than 0.01 or who are missing in pos.txt will be removed. The EM results will be saved. The output filenames begin with "pref." The test statistics (Bayes factors) will be computed after controlling for some covariates detailed in cov.txt file.

3. **Use saved EM parameters.**

   ```
   ./hapqtl -g example/geno.txt -p 1 -pos example/pos.txt   \
   -o pref -C 3 -c 10 -mg 200 -exclude-maf 0.01 --exclude-nopos  \
   -rem output/pref.em.txt -w 1 -FILE example/pheno.txt -DOC example/cov.txt
   ```

   This command line will read EM parameters saved in the previous example, run 1 more EM step, and compute test statistics. Note when reusing saved EM parameters, -C and -c should use the same values that generate the EM parameters. In addition, the SNP filtering options should also be kept the same to guarantee the number of SNPs stays the same.

6 Output files

hapQTL will create output files in a directory named output/. If this directory does not exist then it will be created. Output files will be produced, each with a name beginning with "prefix" that was specified by the -o option. We now describe the contents of these output files.

6.1 Log file: `prefix.log`

A log file includes details of the run parameters used and any warnings generated. When sending in a bug report, it is important to include the log file as an attachment.

6.2 SNP information file: `prefix.snpinfo.txt`

This file contains `6` columns: the SNP rsID, minor allele, major allele, minor allele frequency, chromosome, and position.

6.3 Bayes factors file `prefix.bf.txt`

This file was generated by default. The file has the same number of rows as the number of SNPs, and the SNPs are in the same order as those in `prefix.snpinfo.txt` file. The file contains four columns with header `bf2, pv2, bf1, pv1`, and they are $\log_{10} BF_{hap}$, $-\log_{10} P_{hap}$, $\log_{10} BF_{snp}$, and $-\log_{10} P_{snp}$, respectively, where BF denotes Bayes factors and P denotes p-values, and the subscript *hap* denotes haplotype method and *snp* denotes single-SNP method. Note that when multiple EM runs were invoked (for example, `-e 5`), the reported $\log_{10}$ BFs are $\log_{10}$ of mean BFs averaged over multiple EM runs, however, the reported $-\log_{10}$ P-values are $\log_{10}$ of minimum P-values over multiple EM runs. (The BF is more reliable than the p-values in this context.)

7 Choice of parameters

For the EM steps (specified with `-w`), a number between 20 and 50 is recommended, and 30 is default. Note `-w` invokes the linear approximation algorithm and is much faster than the quadratic algorithm which can be invoked by `-s`. And the two options have similar power. Note, however, because of the approximation, the likelihood of EM steps, if invoked by `-w`, is not strictly increasing, and may oscillate damping towards the point of convergence. For the upper layer number of clusters (specified with -C), 2 (default) or 3 seems working well. For a case/control design, 2 is recommended. For the lower layer number of clusters (specified with -c), 10 is recommended and is the default value.

Appendix A: `hapQTL` Options

Unless otherwise stated, *arg* stands for a string, *num* stands for a number.
File I/O related options:

- -g *arg* can use multiple times, must pair with -p.
- -p *arg* can use multiple times, must pair with -g. *arg* takes either integer values or a string.
- -pos *arg* can use multiple times. *arg* is a file name.
- -o *arg* *arg* will be the prefix of all output files, the random seed will be used by default.
- -FILE *arg* specify a phenotype file.
- -DOC *arg* specify a file containing covariates to be controlled for.

EM Parameters:

- -e *num* specify number of EM runs; default value 1.
- -w *num* specify steps in EM run using linear approximation, default 30.
- -s *num* specify steps in EM run using quadratic method, default 0.
- -C *num* specify number of upper clusters; default value 2.
- -c *num* specify number of lower clusters; default 10.
- -mg *num* specify number of mixture generations; default 100.
- -R *num* specify random seed; use system time by default.
- -sem *num* save EM results to prefix.em.txt.
- -rem *file* read EM from a file.

Other options:

- -v(ver) print version and citation
- -h(help) print this help
- -exclude-maf *num* exclude SNPs whose maf is less than *num* , default 0.
- `--exclude-nopos` exclude SNPs that has no position information
- `--exclude-miss1` exclude SNPs that are missing in at least one file.
- `--silence` no terminal output.

Appendix B: `hapQTL` source code

If you want to compile an executable from the source code, the first thing to do is to install an gsl library, which can be obtained from `http://www.gnu.org/software/gsl/`. Remember the path to which the gsl is installed and modify the Makefile, the one in the src directory, substituting the old path with the correct path. Then you may type make to compile.