Rolf Nevanlinna Institute

Multimapper Reference Manual

Mikko J. Sillanpää

Bayesian QTL mapping software for inbred linecrosses (BC, F2)

Initial version August 11th, 2004

This software is a computer implementation of the Bayesian QTL mapping method that was presented in the paper "Bayesian mapping of multiple quantitative trait loci from incomplete inbred linecross data" by Mikko J. Sillanpää and Elja Arjas (1998); see Sillanpää et al. (2004) for important comment. The program implements the Metropolis-Hastings-Green (Metropolis et al. 1953, Hastings 1970, Green 1995) algorithm in estimation of the model parameters (see the above paper for further detail). The software is written in C-language and is designed for Unix or Linux environment but may work also in other environments.

Please address all the correspondence to:

Mikko J. Sillanpää
Rolf Nevanlinna Institute,
Department of Mathematics and Statistics,
P.O. Box 68, FIN-00014 University of Helsinki, Finland

email : `mjs@rolf.helsinki.fi` web : http://www.rni.helsinki.fi/∼mjs

Technical details

You can "unpack" the file `MM.tar.Z` with the commands:

```
> uncompress MM.tar.Z
> tar -xvf MM.tar
```

The file named `data_structure.h` includes following four lines for Sun UltraSparc workstations:

```
long lrand48(void);
#define rand() lrand48()
#undef RAND_MAX
#define RAND_MAX 2147483647
```

These lines should be commented out when the software is compiled in some Linux environments or in such Unix systems which have different RAND_MAX values. If these lines are omitted in Sun UltraSparc workstation or used in Linux, the software may not work properly and can give wrong results. Before compilation, please check what is a proper RAND_MAX value in your system. The software is compiled with a command **make**. One may need to edit the `Makefile` first. Files `MMfreq.c` and `MMndist.c` are compiled separately.

Running Multimapper Under Windows

As was pointed out to me by Dr. Peter Baker from Australia, one can use **cygwin** (available at http://cygwin.com/) to get gcc, make and other Unix utilities under Windows. By using cygwin gcc, one should be able to compile and run Multimapper under Windows. To speed up the cygwin gcc downloading time, there are some mirror sites which are listed in (http://cygwin.com/mirrors.html) (hopefully they are also up to date).

Registration & Mailing list

No official registration for the execution of the program is required, however, one may inform me that one is willing to joint for the mailing list. It would be appreciate if one would sent me an e-mail message subjected as 'participation to Multimapper mailing list' where she/he would indicate his/her name, institution and e-mail address. I will update given names to the mailing list. This way I could notify the people in the list about a newer version of the program and possible errors in the program. (One may find the same information from my web-page as well.) This information would also give me some feedback how many persons are using the program or are interested in it.

Genetic map and marker data

The input files for the genetic linkage map <*filename*>.*map*, and offspring data <*filename*>.*cro* are the same as those for QTL Cartografer (Basten et al. 1996). The data, which could come from controlled experiment or a simulation, will consist of markers, traits (and other explanatory variables; not supported by Multimapper). The details of the file formats are well documented in the QTL Cartografer manual and thus not presented here. Users are acquaint themselves with these formats by reading the appropriate sections of the QTL Cartografer manual, which is available at no charge from Noth Carolina State University via anonymous ftp at statgen.ncsu.edu. Any questions conserning QTL Cartografer should be directed to Dr. Christopher J. Basten (basten@statgen.ncsu.edu).

Similar file format in Multimapper and in QTL Cartografer has several advantages:

(1) All simulated data sets which have been generated with QTL Cartografer software are directly applicable in Multimapper.
(2) Genetic linkage maps constructed under MAPMAKER/EXP program are applicable in Multimapper after file conversion with QTL Cartografer package.
(3) Data sets in QTL Cartografer format, and data sets in MAPMAKER/QTL format (after

QTL Cartografer conversion) are applicaple in Multimapper.

(4) Many QTL Cartografer support programs (e.g., Qstats) can also be utilized.

Multimapper directive files

Four directive files are needed in your working directory to be able to run the Multimapper program. These files are :

(1) **randomwalk.file** (2) **bg_controls.file**

(3) **priorlimits.file** (4) **codesystem.file**

Contents and formats of the files are presented in the following four sections.

1 randomwalk.file

In this file, the user defines the chromosome and the trait of interest; the number of MCMC rounds, and the chromosomal starting points. Proposal ranges which directly affect the rejection rates are also specified here as well as the design of the data.

The randomwalk.file has a following format:

1. a chromosome number (controls which chromosome is being mapped).
2. a trait number (controls which trait is being mapped).
3. the number of MCMC rounds to be run.
4. initial locations of the three QTLs on the chromosome in centiMorgans (cM). Only restriction being that they must be separate points on the prior range ($\leq$ length of the chromosome).
5. $\frac{1}{p_a}$ $(= \frac{1}{p_d})$, where p_a (p_d) is a proposal probability to add (delete) a QTL.
6. a (common) proposal range of the QTL location parameters.
7. a proposal range of the regression mean (a).
8. a proposal range of the residual variance (σ^2).
9. a proposal range of the regression coefficients of QTL genotypes.

10. a proposal range of the regression coefficients of the background control genotypes.

11. {1 or 0 }, where 1 indicates that regression coefficients are printed to a output file.

12. the data design { 1= backcross, 2= F2 }

an example file looks like this :

```
12345678901234567890123456789 <- column number
xxxxxxxxx xxxxxxxxx xxxxxxxxx <- indicates length of the fields
1                             < - file starts from this line
1
500000
20.0      50.0      80.0
60.0
 1.0
 0.01
 0.01
 0.10
 0.10
1
1                             < - file ends at this line
```

2 bg_control.file

The background controls for your data are assumed to be defined in some preliminary analyses (e.g. stepwise regression) and are introduced in this file as known entities.

In this file must include as many rows as there are background controls, representing one line to each background control. There are two parameters in each line: the first parameter is the chromosome number and second parameter is the marker number. The first parameter (column) indicates on which chromosome the current background control locates. Second parameter is a marker number (bc) representing the closest location of a particular QTL on a given chromosome. Numbering of the chromosomes and markers are expected to start from one.

At the very beginning the program asks how many background controls are needed and reads given number of bc from the start of `bg_control.file`.

An example of `bg_controls.file` containing 5 background controls:

```
1234567890
xxxx xxxx
   1    2      < - file starts from this line
   2    3
   2    5
   4    4
   5    2      < - file ends at this line
```

3 priorlimits.file

The prior limits for the regression parameters are specified in this file. The priorlimits specify the the parameter space where the parameter values are acceptable. Common priorlimits for all genotypic regression coefficients are assumed. Also common limits for all background control coefficients are assumed.

The priorlimits.file includes always just 5 lines specifying following information in following order:

1. lower and upper limits for the regression intercept parameter (a).
2. lower and upper limits for the residual variance (σ^2).
3. lower and upper limits for the regression coefficients of the QTL genotypes
4. lower and upper limits for the regression coefficients of the background control genotypes
5. lower and upper limits for the QTL location(s) specifying chromosomal segment to be mapped.

Example of `priorlimits.file`:

```
1234567890123456789
xxxxxxxxx xxxxxxxxx
  -1000.0    1000.0      < - file starts from this line
      0.0     170.0
   -500.0     500.0
   -500.0     500.0
      0.0     100.0      < - and ends to this
```

Note that a natural lower bound for σ^2 is zero and upper bound is the phenotypic variance. Note

also that if given upper bound for the QTL-location is bigger than the chromosomal length, the program automatically uses the chromosomal length as an upper bound.

4 codesystem.file

This file is designed to cover more general data designs than BC or F2 from inbred line cross. However, in current version of the program, only BC and F2 are supported in other modules.

All possible genotypes of the design must be presented in this file.

How the genotype are coded can be presented in the form :

homozygotes : 1, ..., N_{homoz}

Heterozygote (ij) 1000*i+j for all (ij) combinations, however in a such way that duplicates (ji)=(ij) are not included.

The first column indicates genotype, the second column indicates genotype-code in the *.cro file, and the third column must be coded as presented above. Example file for a design with 4 possible homozygotes and total of 10 possible genotypes, is presented in the following format. Note that the partial genotype info must appear after the complete genotype info.

An example of codesystem.file :

```
12345678901234
xx xxxx xxxxxx
AA   1     1      < - file starts from this line
BB   2     2
CC   3     3
DD   4     4
AB   5    1002
AC   6    1003
AD   7    1004
BC   8    2003
BD   9    2004
CD  10    3004
A-  -2   -1       < - partial genotype including A-allele
B-  -3   -2       < - partial genotype including B-allele
C-  -4   -3       < - partial genotype including C-allele
D-  -5   -4       < - partial genotype including D-allele
--  -1   -1000    < - list of genotypes always end to info from missing
```

```
                    genotype, which is also indicator for the end of file.

Example for a backcross :
AA   1     1
AB   2    1002
--  -1   -1000


and F2 intercross :
AA   1     1
BB   3     2
AB   2    1002
--  -1   -1000
```

Timestamps

When the execution of the Multimapper is started, the program creates a new file named `MH_timestamp`. After finishing, you can check how much real computing time was required by calculating the difference between the last updating time and the creation time of the `MH_timestamp` file that can be found from a directory. Note that the time calculated in this way, is relative to the other load in the computer. However, when there is no other load in the computer, the execution time is accurate.

Program outputs

1) Technical notes of the run are collected into a file called `MH_output.logi`.

2) In each MCMC round, the program prints out a value of the variable N_{qtl} (number of QTLs) into a ASCII file named `MH_output__Nqtl`.

3) The program prints out QTL locations from MCMC rounds into three files: `MH_output__lx0`, `MH_output__lx1`, and `MH_output__lx2`. If the number of QTLs in the MCMC round is zero no files are updated, if it is one, only first file is updated, if it is two, two files are updated and so on.

4) Similarly, program prints out QTL genotypic coefficients into three files :`MH_output__reg0`, `MH_output__reg1`, and `MH_output__reg2`.

5) The program prints out a value of residual variance (σ^2) and of intercept a (non-identifiable) into a file named `MH_output__reg_sigma_a`. First column is σ^2.

Estimating posterior QTL-intensity from MCMC output

Summarizing posterior density of QTL (or posterior QTL-intensity) based on information on large MCMC sample is important final task. See Hoti et al. (2002) for kernel density estimation of QTL-intensity and general discussion on this topic. Kernel density estimation outperforms histogram approximation and therefore should be used in general, especially when estimating mode (i.e. the best putative QTL position on the marker map). Different kernel implementations (Matlab-programs) to summarize QTL position information are available at Fabian Hoti's homepage (http://www.rni.helsinki.fi/~fjh).

Construction of simple histogram approximation for posterior QTL intensity from large MCMC output files can be easily done with C-program named `MMfreq`. It produces output file (output.txt) which can be easily visualised with any visualisation program. This program does not require lot of memory for execution and it can be applied for compressed files.

Program `MMfreq` reads its directives from standard input and can be executed in following way :

```
cat MH_output__lx* | MMfreq LC NC output.txt
```

Alternatively, if the output-files are in a compressed form:

```
zcat MH_output__lx* | MMfreq LC NC output.txt
```

LC = Lenght of the analysed chromosome (linkage group)

NC = number of resulting histogram - classes (i.e., bins)

$\Rightarrow$ width of each class is then obtained as LC/NC.

output.txt = name of the output-file in where frequencies are stored in the format:

(bar center, frequency) separated with space.

Posterior prob. distribution for different number of QTLs

In the package, there is a C-program named `MMndist` which reads file `MH_output__Nqtl` and prints out four (approximative) posterior probabilities of (number of QTLs) N_{qtl} having value 0,1,2 or 3 and also (approximative) posterior expectation for N_{qtl}. Note that the prior distribution assumed for number of QTLs in the method (and the software) is actually an accelerated truncated Poisson distribution instead of ordinary Poisson distribution (see Sillanpää et al. 2004).

Matlab

Visualisation of and construction of the QTL intensity and the phenotypic effect graphs from MCMC output files, can easily be created with Matlab software. I have used version 4.2c. Some example scripts that can be run under Matlab are found below.

Plot of sample path for the number of QTLs:

```
load MH_output__Nqtl
plot(MH_output__Nqtl)
```

plotting QTL-intensity histogram constructed by MMfreq :

```
load output.txt
x=output(:,1);
```

```
bar(x,output(:,2));
```

construction and plotting QTL-intensity histogram :

```
load MH_output__lx0
load MH_output__lx1
load MH_output__lx2


k=[ MH_output__lx0
    MH_output__lx1
    MH_output__lx2];

clear MH_output__lx0 MH_output__lx1 MH_output__lx2;

bin=zeros(1,100);
for s=1:100
bin(s)=(s-0.5)/100;
end

[n,x]=hist(k,bin);


save chromo.mat n x

bar(x,n);
```

If one is only interested in a QTL-intensity histogram, for a shortcut, one may write after the clear-sentence just one line : `hist(k,100)`

and a graph of phenotypic effects:

```
load MH_output__lx0
load MH_output__lx1
load MH_output__lx2

load MH_output__reg0
load MH_output__reg1
load MH_output__reg2


k=[MH_output__lx0
   MH_output__lx1
   MH_output__lx2];

kd=[MH_output__reg0(:,2)-MH_output__reg0(:,1)
    MH_output__reg1(:,2)-MH_output__reg1(:,1)
    MH_output__reg2(:,2)-MH_output__reg2(:,1)];

clear MH_output__lx0 MH_output__lx1 MH_output__lx2;
clear MH_output__reg0 MH_output__reg1 MH_output__reg2;

N=length(k);
load chromo.mat
```

```
lkm=length(n);
m=zeros(lkm,3);
ug=zeros(lkm,3);
dq=zeros(lkm,3);
pp=zeros(1,lkm);

xl=x;
p=zeros(1,lkm);

disp('classification...');
for s=1:N
if rem(s,1000) == 0
disp(s)
end

[dummy,l]=min(abs(xl-k(s)));
k(s) = l(1);

end
disp('..classification');

disp('median and quantiles..');
for l=1:lkm
pist=find(k==l);
p=length(pist);
pp(l)=p;
y=sort(kd(pist));
if ~isempty(y)
m(l)=median(y);
ind2=floor(0.025*p);
if ind2==0
ind2=1
end
uq(l)=y(ceil(0.975*p));
dq(l)=y(ind2);
else
m(l)=0.0;
uq(l)=0.0;
dq(l)=0.0;
end

end

disp('done');
end

axis([0,100,-2.0,1.5])

plot(xl,m);

hold on

plot(xl,uq,'w:');

hold on

plot(xl,dq,'w:');
```

Execution in UNIX/LINUX by using script file

It is possible to execute the program with a command: runthis > output.out &

where runthis is a following executable script file :

```
#!/bin/sh
main <<END
simul.map
simul.cro
5
END
```

simul.map (simul.cro) is a given filename for genetic linkage map (data) and 5 is a given number of background controls.

Acknowledgement

M.S. is grateful for Matti Taskinen for providing the Makefile, MMfreq and executable script-file for the software, and for Dr. Cristopher J. Basten allowing to use QTL Cartografer *.map and *.cro fileformats in this software.

References

Basten, C. J., B. S. Weir, and Z.-B. Zeng (1996) QTL Cartografer, the reference manual and tutorial for QTL mapping. North Carolina State University anonymous ftp: esssjp.stat.ncsu.edu (in the directory /pub/qtlcart).

Green, P. J. (1995) Reversible jump Markov Chain Monte Carlo computation and Bayesian model determination. Biometrika **82**: 711-732.

Hastings, W. K. (1970) Monte Carlo sampling methods using Markov chains and their applications. Biometrika **57**: 97-109.

Hoti, F. J., M. J. Sillanpää, and L. Holmström (2002) A note on estimating the posterior density of a quantitative trait locus from a Markov chain Monte Carlo sample. Genetic Epidemiology **22**: 369-376.

Metropolis, N., A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller (1953) Equation of state calculations by fast computing machines. Journal of Chemical Physics **21**: 1087-1092.

Sillanpää, M. J. and E. Arjas (1998) Bayesian mapping of multiple quantitative trait loci from incomplete inbred line cross data. Genetics **148**: 1373-1388.

Sillanpää, M. J., D. Gasbarra, and E. Arjas (2004) Comment on the "On the Metropolis-Hastings acceptance probability to add or drop a quantitative trait locus in Markov chain Monte Carlo-based Bayesian analyses". Genetics **167**: 1037.