
Getting Started with HapCluster

An introduction to the HapCluster association mapping tool

Thomas Mailund
mailund@stats.ox.ac.uk

Copyright © 2006 Thomas Mailund • Dept. of Statistics, University of Oxford,
United Kingdom

Permission is granted to make and distribute verbatim copies of this manual
provided the copyright notice and this permission notice are preserved in all
copies.

About HapCluster

HapCluster++ is a software package for linkage disequilibrium mapping. It is based on a Bayesian Markov-chain Monte Carlo (MCMC) method for fine-scale linkage-disequilibrium gene mapping using high-density marker maps. HapCluster++ is a C++ implementation of the method described in the paper

Fine Mapping of Disease Genes via Haplotype Clustering. E.R.B. Waldron, J.C. Whitaker and D.J. Balding. *Genetic Epidemiology*. 30: 170–179. (2006)

HapCluster is released under the [GNU General Public License \(GPL\)](#). There are no restrictions on the use of HapCluster, for commercial or academic use, but the use of the source code, in part or in whole, is restricted according to the GPL.

Installing HapCluster

HapCluster is distributed as RPM files or as source code. For most users, we recommend installing from the RPM files, since building the tool from source requires setting up the right build environment and having access to the needed development tools. If you are not familiar with UNIX C++ development—using the Automake suite of tools—we do not recommend that you try building from source.

Installing the RPM Files. The RPM file, `hapcluster-x.y.z-r.i386.rpm`—where `x.y.z-r` is the version and release number—contains a binary version of HapCluster, compiled to an Intel x86 Linux platform. To install HapCluster from the RPM package, run

```
sh> rpm -Uvh hapcluster-x.y.z-r.i386.rpm
```

Since the RPM files installs in the directory `/usr/local/`, installing the RPM package requires root access.

Installing from the Source Files. The source code is distributed in a tar-file, `hapcluster-x.y.z.tar.gz`. To build the source files, first uncompress and untar the file, then run ‘configure’ and finally ‘make’. To test that the build was successful, run ‘make check’. To install the program, run ‘make install’.

```
sh> tar zxf hapcluster-x.y.z.tar.gz
sh> cd hapcluster-x.y.z
sh> ./configure
sh> make
sh> make check
sh> make install
```

Using HapCluster

Analysing a single genomic region

For running HapCluster, you need to specify: 1) A list of marker positions, and 2) A list of haplotypes—containing a SNP value (0 or 1) for each marker—together with case/control status.

```
sh> hapcluster positions.txt haplotypes.txt
```

The positions file should contain a list of numbers, sorted ascending. The format of the haplotype file is: One line per haplotype, where a haplotype is represented as a list of space-separated alleles, and each allele represented as either a '0' or a '1'. The first column is a 'pseudo'-allele used for the case/control dichotomy: a '0' in the first column is taken to mean that the haplotype is a *control* haplotype and a '1' at the first column is taken to mean that the haplotype is a *case* haplotype.

Optionally, the number of MCMC iterations, the number of burn-in iterations, and the thinning between samples can be specified. It is also recommended that you specify the maximal window size for regions considered in the clustering distance score (see *Waldron et al. 2006* for details). To see a list of all supported options, use the option `-h` or `--help`:

```
sh> hapcluster -h
```

The output will be written to the terminal—or, if the `-o` or `--output` option is used, to a file—and will contain a table with a row for each sample and a column for each sampled value, including the log-likelihood, l , the disease locus, x , the cluster distance, δ , mismatch penalty γ , the acceptance rate of the MCMC and the various counts for the number of each genotype implied by the clustering (see *Waldron et al. 2006* for details).

This output can then be analysed in e.g. R (<http://www.r-project.org>). If, for example, we run HapCluster and write the samples to the file `samples.txt`:¹

```
sh> hapcluster -v -o samples.txt \  
CYP-positions.txt CYP-haplotypes.txt
```

reading this file into R we can plot the likelihood

```
> samples <- read.table('samples.txt', header=TRUE)  
> plot(samples[, 'likelihood'], type='l',  
+       main='Likelihood', ylab='log-likelihood')
```

(see Fig. 1) or the disease locus samples

```
> plot(samples[, 'x'], type='l',  
+       main='Locus', ylab='Disease Locus')
```

(see Fig. 2).

¹The data used in this example is the CYP2D6 study described in *Hosking et al.* Linkage disequilibrium mapping identifies a 390 kb region associated with CYP2D6 poor drug metabolising activity, *J. Pharmacogenomics* 2002.

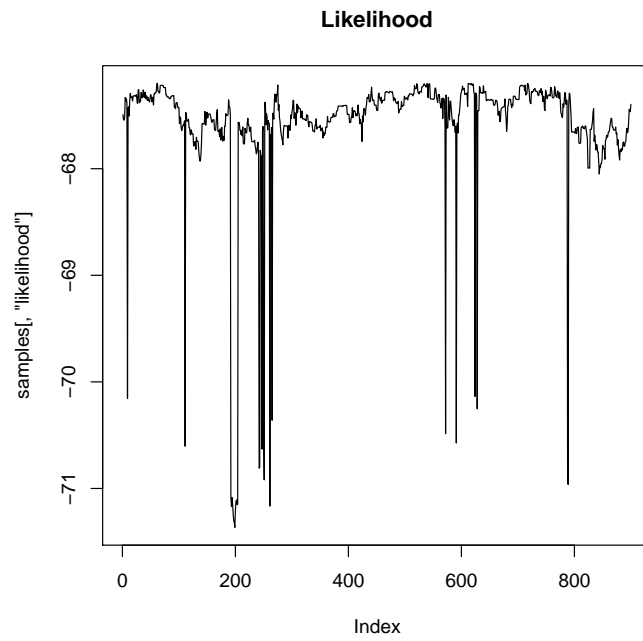


Figure 1: Plot of the sampled likelihood.

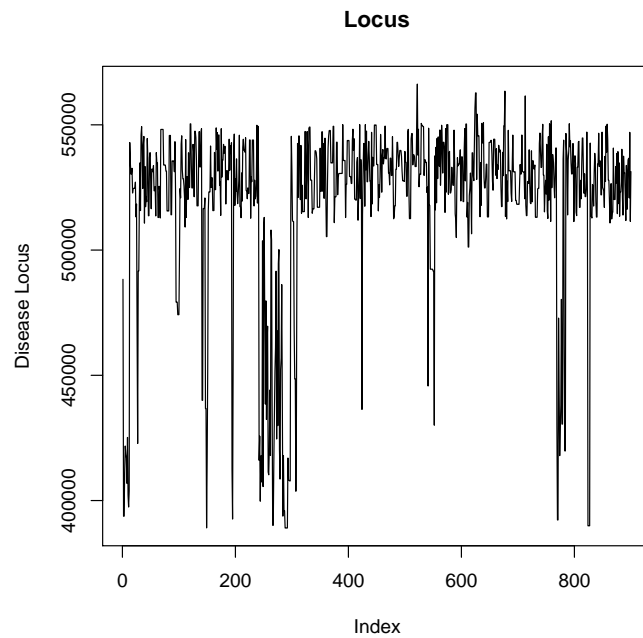


Figure 2: Plot of the sampled disease locus.

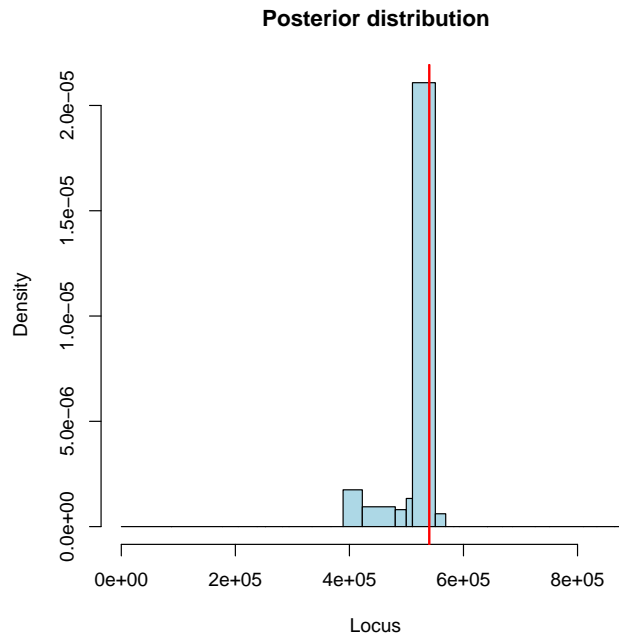


Figure 3: Posterior distribution of the CYP dataset. The red vertical line indicates the true disease locus.

We can plot the posterior distribution as a histogram using

```
> hist(samples[, 'x'], breaks=scan('CYP-positions.txt'),
+       main='Posterior distribution', xlab='Locus',
+       col='lightblue')
> abline(v=540000, col='red', lwd=2)
```

(see Fig. 3) where we use as break-points the marker positions (all loci between the same two markers will have the same likelihood, so the marker-intervals is the finest resolution the tool will pick up, see *Waldron et al.* 2006 for details).

Analysing unlinked, interacting regions

HapCluster also supports an experimental method for mapping in the presence of interaction between two (unliked) genes. To map with interaction, use the program `ihapcluster` and specify two input regions (in the form of a position file followed by a haplotype file). Use option `-h` or `--help` to get a list of supported options for `ihapcluster`.

```
sh> ihapcluster positions.1.txt haplotypes.1.txt \
           positions.2.txt haplotypes.2.txt
```

The output will contain the joint distribution of disease loci in the columns `x1` and `x2`.

Contact

For comments or questions regarding HapCluster, please contact Thomas Mailund (mailund@stats.ox.ac.uk).