HapSeq – A Program for Genotype Calling and Haplotype Phasing from Next Generation Sequencing Data Using Haplotype Information of Reads

Degui Zhi and Kui Zhang

January 4, 2013

This document describes how to use HapSeq, a program for genotyping calling and haplotype phasing from next generation sequencing data using haplotype information from jumping reads. We developed a Hidden Markov Model (HMM) based method for genotype calling and haplotype phasing from next generation data that can take into account jumping reads information across two adjacent potential polymorphic sites. Our method extends the HMM in the Thunder program (Li, et al., 2010) and explicitly models jumping reads information as emission probabilities conditional on the states of adjacent sites. The method is implemented in the program, HapSeq. The program is written with C++ and based on the source code of Thunder program provided by Drs. Yun Li and Goncalo Abecasis. For the detailed description of the method implemented in HapSeq, please refer to our manuscript (Zhi et al., 2012).

Program Log

December 12, 2011

- The executable version of HapSeq is implemented and released.

January 10, 2013

- The manual and web site are updated.

Compile the Program

To run HapSeq, you need a compiled copy of program. At this time, we only provide the compiled program under the Windows XP system and Linux system. Please contact us if you need the compiled program for other operating systems.

Execute the Program – the Command Line and the Options

The program runs under a command line. The following command line:

```
./HapSeq --readCounts counts1.txt --polymorphicSites sites1.txt
--readHap jumps1.txt --seqError 0.01 --rounds 100 --seed 10
--phase --geno --quality --uncompressed --prefix res-hapseq-r10-1
```

shows a the simple usage of HapSeq. We will explain these options in detail in the subsequent sections.

Input Files

HapSeq uses three input files: the count file, the site file, and the haplotype count (“jump”) file from jumping reads that cover two consecutive potential polymorphic sites. The count file and the site file are required and have the same format as those in Thunder. The haplotype count file is optional. If the haplotype count file is absent, HaqSeq has the same behavior as thunder.

Input File – the Site File

The site file is the text file and is set by the option “--polymorphicSites“. This file contains the information of sites. The first few rows look like this:

```
S1    1     2
S2    1     2
S3    1     2
S4    1     2
S5    1     2
```

Each row represents the information of a bi-allelic site and the number of rows in the site file is the number of sites used in HapSeq. There are three columns for each row which are separated by the space or tab: the first column is the name of site, the second and third columns are two alleles at that site. In the current implementation of HapSeq, only bi-allelic sites can be used.

Input File – the Count File

The count file is the text file and is set by the option "--readCounts". This file contains count data of two alleles at each site for each individual. The first few rows and columns look like this:

1	1	0	0	1	0	7	0	3	…
2	2	0	0	1	0	0	0	4	…
3	3	0	0	1	2	4	0	2	…
4	4	0	0	1	0	1	0	6	…
5	5	0	0	1	0	6	0	1	…
6	6	0	0	1	5	1	0	2	…

Each row repents the information of an individual and the number of rows in the count file is the number of individuals used in HapSeq. The number of columns of each row equals the summation of 2 and 2 times the number of sites in the site file. The first five columns represent the family id, the individual id, the father id, the mother id, and the gender of that individual. For the subsequent columns, each pair of columns represents the read counts for allele 1 and allele 2 at that site, respectively. Since HapSeq can only handles unrelated individuals at this moment, the father id and the mother id should be 0 in the count file.

Input File – the Haplotype Count File

The haplotype count file is the text file and is set by the option "--readHap". This file contains the information of sequencing reads that cover two adjacent for each individual. The first few rows look like this:

1	1	0	0	1	170	171	0	2	1	0
1	1	0	0	1	370	371	0	0	0	5
1	1	0	0	1	70	71	0	3	0	2
1	1	0	0	1	119	120	0	0	1	2
1	1	0	0	1	166	167	0	0	1	0
1	1	0	0	1	88	89	0	0	0	1
1	1	0	0	1	222	223	0	0	0	2

Each row repents the haplotype counts of two adjacent sites of an individual. The number of columns of each row is 11. The first five columns, which are same as those in the count file, represent the family id, the individual id, the father id, the mother id, and the gender of that individual. The 6th and 7th columns are the index of two adjacent sites covered by the jumping reads. The index starts from 1 so the first site in the site file is indexed as 1, and the second site in the site file is indexed as 2, etc. The current implementation of HapSeq can only handle haplotype counts from two adjacent sites therefore the two indexes are two consecutive positive integers. This file can also include haplotype counts from two non adjacent sites but such haplotype counts will be ignored by HapSeq. The last four columns represent the four haplotype counts of 11, 12, 21, and 22 from jumping reads, respectively.

How to Prepare the Count and Haplotype Count File

The count and haplotype count should be obtained from read alignment (BAM) files. The count file contains the read counts of two alleles at each site from all reads including jumping reads while the haplotype count file only contains the number of haplotype counts from jumping reads. If the haplotype count file is supplied, HapSeq will use the modified counts data according to the haplotype counts. Otherwise HapSeq uses the counts data from the file and has the same results with Thunder. We include two Perl scripts that can generate these files from a set of BAM files:

- hapseq_pipeline.pl: script to prepare count and haplotype count data from a set of BAM files.
- bam_to_counts.pl: script to parse a BAM file and generate counts and jumps file for one individual. This script is called from hapseq_pipeline.pl.

Output Files

There are two main output files: the imputed genotypes at each site for each individual and the inferred pair of haplotypes for each individual. These files have the same format as those from Thunder. We have included two example files.

Other Options

Since HapSeq was implemented based on Thunder it shares all options that can be used by Thunder. The users can refer to Thunder for more details. Here we list some other important options that should be configured by the users in **Table 1**. Notice (1) each option is led by "--" and there is a space between the option and its argument. (2) Two options, "--rounds" and "--prefix" have the short version "-r" and "-o", respectively. The short version of option is led by "-" and there is still a space between the option and its argument.

Table 1: The options used in HapSeq.

Option	Argument	Description	Default Value
--readCounts	A character string	The count file	No default, must be specified
--polymorphicSites	A character string	The site file	No default, must be specified
--readHap	A character string	The haplotype count file	No default
--seed	A positive integer	The random seeds	123456
--seqError	A real number	The error rate for sequencing data	0.005
--burnin	An non-negative integer	The number of burn in for the HMM	0
--rounds	An positive integer	The number of iteration for sampling	Suggest to use at least 50
--phase	No argument	If output haplotypes	
--geno	No argument	If impute genotypes	
--prefix	A character string	The prefix for output files	No default, must be specified

Examples

Here we provide an example with a count file (counts1.txt), a site file (sites1.txt), a haplotype count file (jumps1.txt). The output files, res-hapseq-r10-1 and res-hapseq-r10-1.geno, were obtained with the following command line:

```
./HapSeq --readCounts counts1.txt --polymorphicSites sites1.txt
--readHap jumps1.txt --seqError 0.01 --rounds 100 --seed 10
--phase --geno --quality --uncompressed --prefix res-hapseq-r10-1
```

The output files, res-thunder-r10-1 and res-thunder-r10-1.geno, were obtained with the following command line without the use of haplotype count file:

```
./HapSeq --readCounts counts1.txt --polymorphicSites sites1.txt
--seqError 0.01 --rounds 100 --seed 10
--phase --geno --quality --uncompressed --prefix res-thunder-r10-1
```

Remarks

In this section, we highlight some important points and/or possible solutions for the problems that may be encountered in running HapSeq.

- When you prepare the haplotype count file, be aware that the index of sites starts from 1.
- Since both HapSeq2 and Thunder are HMM based, the different random seeds from different runs will generate different results. The combined results from multiple runs with different random seeds are generally more accurate that results from a single run.

Contact Information

This program and related materials can be downloaded through the following web site:

http://www.ssg.uab.edu/hapseq

Bugs, comments, or the request of compiled program with other operating systems should be reported to:

Degui Zhi
Department of Biostatistics
University of Alabama at Birmingham
Ryals Public Health Bldg. 327L
1665 University Blvd., Birmingham, AL, 35294
Phone: 205-975-9192
Fax: 205-975-2540
Email: dzhi@uab.edu

Kui Zhang
Department of Biostatistics
University of Alabama at Birmingham
Ryals Public Health Bldg. 327H
1665 University Blvd., Birmingham, AL, 35294
Phone: 205-996-4094
Fax: 205-975-2540
Email: kzhang@uab.edu

References

Yun Li, et al. 2010. MaCH: using sequence and genotype data to estimate haplotypes and unobserved genotypes. *Genetic epidemiology*, 34, 816-834.

Zhi D, Wu J, Liu N, Zhang K. 2012. Genotype calling from next generation sequencing data using haplotype information of reads. *Bioinformatics* 28: 938-946.