HapSeq2 – A Program for Haplotype Phasing and Genotype Calling from Next Generation Sequencing Data Using Haplotype Information of Reads

Degui Zhi and Kui Zhang

June 5, 2013

This document describes how to use HapSeq2, the updated version of HapSeq. HapSeq2 is a program for genotyping calling and haplotype phasing from next generation sequencing data using haplotype information from jumping reads. Previously, 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 (Zhi et al., 2012). 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).

Recently, we extended our methods to incorporate the haplotype information of multiple adjacent and/or non-adjacent sites from sequencing reads. Our model is inspired by the model implemented in HASH that was originally designed to phase individual genomes (Bansal et al., 2008). We develop a new hybrid MCMC algorithm that combines the Gibbs sampling algorithm of HapSeq and Metropolis-Hastings algorithm similar to that of HASH and is computationally feasible. We show by simulations and real data from the 1000 Genomes Project that our model offers superior performance for haplotype phasing as well as genotype calling for population NGS data over existing methods. The new method is implemented in this program, HapSeq2. For the detailed description of the method implemented in HapSeq2, please refer to our manuscripts (Zhi et al., 2012; Zhang and Zhi, 2013).

Program Log

December 12, 2011

- The executable version of HapSeq (HapSeq1) is implemented and released.

January 4, 2013

- The executable version of HapSeq2 is implemented and released.

June 5, 2013

- The updated version of HapSeq2 with several refined options is implemented and released.

Compile the Program

To run HapSeq2, you need a compiled copy of program. At this time, we only provide the compiled program under the Windows XP and Linux operating 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:

```
./hapseq2 --readCounts count.txt --polymorphicSites sites.txt
--readHap jump1.txt --readForwOpt 1
--seqReadFile read.txt --mhPhasing --mhRounds 05
--seed 10 --seqError 0.01 --rounds 100 –o res-hapseq
--phase --geno --quality
```

shows a the simple usage of HapSeq2. We will explain these options in detail in the subsequent sections.

Input Files

HapSeq2 uses four input files: the count file, the site file, and the haplotype count ("jump") file from jumping reads that cover two consecutive potential polymorphic sites, and the sequencing read file that contains the sequencing reads that cover two or more adjacent and non-adjacent sites. The count file and the site file are required and have the same format as those in Thunder. The haplotype count file and the sequencing read file are optional. If only the haplotype count file is available, HapSeq2 is same as the old version of HapSeq (HapSeq1). If both the haplotype count file and the sequencing read file are absent, HaqSeq2 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 HapSeq2. 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 HapSeq2, 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 HapSeq2. The number of columns of each row equals the summation of 5 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 HapSeq2 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 sites 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 HapSeq2 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 HapSeq2. The last four columns represent the four haplotype counts of 11, 12, 21, and 22 from jumping reads, respectively.

Input File – the Sequencing Read File

The sequencing read file is the text file and is set by the option "--seqReadFile". This file contains the information of sequencing reads that cover two and more adjacent and non-adjacent sites. The first few rows look like this:

```
1  1  0  0  1  64  3  15  16  20   0  1  0
1  1  0  0  1  65  3  15  16  17   0  1  0
1  1  0  0  1  66  2  15  16   0   1
1  1  0  0  1  67  1  16   1
1  1  0  0  1  68  1  16   1
1  1  0  0  1  69  4  16  17  18  19  1  0  0  0
1  1  0  0  1  70  2  16  20   1   0
```

Each row repents a read that covers two and more adjacent and non-adjacent sites. Due to the variant number of sites covered by a read, the number of elements of each row differs. 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 column is the index of this read and is actually not by the program. The 7th column represents the number of sites covered by the read. The rest of columns are divided into two parts: the first part contain the index of sites covered by this read. 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 second part contains the observed allele of this read at corresponding sites. Using the first row as an example, we know that this read is from the individual 1 of family 1. The read covers three sites: sites 15, 16 and 20 (paired-end read). The observed alleles at the sites 15, 16 and 20 are 0, 1, and 0, respectively.

How to Prepare the Count, Haplotype Count, Sequencing Read File

The count, haplotype count, and sequencing read file 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. The sequencing read file contains all reads that cover two and more adjacent and non-adjacent sites. We have implemented several programs to extract the input files required by HapSeq2. The programs and the pipeline to use these programs are described at the following web site: http://www.ssg.uab.edu/wiki/x/RoAPAQ.

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 HapSeq2 was implemented based on HapSeq/Thunder it shares all options that can be used by HapSeq/Thunder. The users can refer to HapSeq/Thunder for more details. Here we list some other important options that should be configured by the users in **Table 1**. Note that: (1) each option is led by "--" and there is a space between the option and its argument. (2) Three options, "--seqError", "--rounds" and "--prefix" have the short version "-e", "-r" and "-o", respectively. The short version of an option is led by "-" and there is still a space between the option and its argument. (3) For an option taking either true or false (the Boolean option), the default value is always false and no argument is needed. If such an option is specified in the command line, its value becomes true.

Table 1: The options used in HapSeq2. The additional options used in HapSeq2 are in red.

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
--readForwOpt	A non-negative integer	If perform the HMM using the haplotype count data 1: perform 0: not perform	0, not perform the HMM with the haplotype count data
--seqReadFile	A character string	The sequencing read file	No default
--mhPhasing	No argument for this Boolean option	If perform the MH sampling using the sequence read file true: perform the MH sampling false: not perform the MH	false, not perform the MH sampling
--mhRounds	A positive integer	The number of rounds for the MH sampling Only effective if --mhPhasing is set as true	5
--mhBurnin	A positive integer	The number of rounds for the MH sampling that is used as burn in Only effective if --mhPhasing is set as true and should be less than --mhRounds	0
--mhWeights	A non-negative integer	The weight used to calculate the probability of proposed haplotype pair in the MH sampling 0: uniform weight 2: the weight used in paper	2

--mhConsensus	No argument for this Boolean option	If the consensus haplotype pair from the MH sampling will be used in the next step of HMM. Note the computation can be heavy when the number of iterations for the MH sampling is large if this option is set as true. true: the consensus haplotype pair will be used	false
--mhLast	No argument for this Boolean option	If the last haplotype pair from the MH sampling will be used in the next step of HMM. true: the last haplotype pair from the MH sampling will be used If both --mhConsensus and --mhLast are set as false then the haplotype pair with the maximum likelihood will be used.	false
--mhDetail	No argument for this Boolean option	If output the details of the MH sampling including the likelihood of the current and proposed haplotype pairs true: output details false: not output details	false
--seed	A positive integer	The random seeds	123456
--seqError **-e**	A real number	The error rate for sequencing data	0.005
--burnin	An non-negative	The number of burn in for the HMM	0

	integer		
--rounds **-r**	An positive integer	The number of iterations for the HMM	Suggest to use at least 50
--phase	No argument for this Boolean option	If output haplotypes true: output haplotype false: not output haplotype	false
--geno	No argument for this Boolean option	If impute and output genotypes true: output genotypes false: not output genotypes	false
--quality	No argument for this Boolean option	If calculate and output quality score true: output false: not output	false
--dosage	No argument for this Boolean option	If calculate and output genotypic dosage true: output false: not output	false
--prefix **-o**	A character string	The prefix for the output files	No default, must be specified
--states	A positive integer	The number of states (reference haplotypes) used in the HMM. Set to a small number to speed up the computation.	2 * the number of samples - 2

Examples

Here we provide an example with a count file (count.txt), a site file (sites.txt), a haplotype count file (jump1.txt), and a sequencing read file (read.txt). The output files, res-hapseq and res-hapseq.geno, were obtained with the following command line:

```
./hapseq2 --readCounts count.txt --polymorphicSites sites.txt
--readHap jump1.txt --readForwOpt 1
--seqReadFile read.txt --mhPhasing --mhRounds 05
--seed 10 --seqError 0.01 --rounds 100 --prefix res-hapseq
--phase --geno --quality
```

Remarks

In this section, we highlight some important points and/or possible solutions for the problems that may be encountered in running HapSeq2.

- When you prepare the haplotype count and sequencing read file, be aware that the index of sites starts from 1.
- Since both HapSeq2 and Thunder are HMM Monte Carlo sampling based, 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.
- For HapSeq2, you need to specify the haplotype count and sequencing read file and specify the options (“--readForwOpt" and “--mhPhasing") to use them. For the old version of HapSeq (HapSeq1) that only takes the haplotype count file, the program uses it once it is specified.
- The computation of the HMM and MH sampling can be intensive. To reduce the computational burden, you can use “--states” to specify a less number of reference haplotypes.

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@ms.soph.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@ms.soph.uab.edu

References

Bansal V, Halpern AL, Axelord N, Bafna V. 2008. An MCMC algorithm for haplotype assembly from whole-genome sequence data. *Genome Research* 18: 1336-1346.

Li Y, Willer CJ, Scheet P, Abecasis GR. 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.

Zhang K, Zhi D. 2013. Joint haplotype phasing and genotype calling of multiple individuals using haplotype informative reads. Submitted *Bioinformatics*.