HotspotFisher

A package for detecting recombination hotspots from population polymorphism data

Last Updated: May 17, 2006.

Jun Li

Bioinformatics Division, TNLIST and Department of Automation, Tsinghua University, China

Address for correspondence: FIT 1-107, Tsinghua University, Beijing 100084, China.

Email: jun-li00@mails.tsinghua.edu.cn

WWW: http://bioinfo.au.tsinghua.edu.cn/member/~lijun

1. Introduction

HotspotFisher is a package for detecting recombination hotspots from population polymorphism data. Written in standard C++, it can be complied and executed in various operating systems, such as Linux/Unix and Windows. HotspotFisher uses a multi-hotspot model and the truncated weighted pairwise log-likelihood (TWPLL), so it can detect multiple hotspots in a region. HotspotFisher can be used to both phased / haplotype and unphased / genotype data directly, with arbitrary levels of missing data.

2. Input Format

HotspotFisher uses population polymorphism data given by two files: a file of SNP locations and a file of haplotypes / genotypes. For users' convenience, the same format as that in LDhat 2.0 is adopted.

For a file of SNP locations, three elements should appear in the first line: (1) number of SNPs, (2) length of the segment, (3) crossover model (L) or gene conversion model (C). In HotspotFisher, the second element does not need to be the real length, since it is not actually used it in the program. The third element must be set as "L" in HotspotFisher. Following the head line are the relative or absolute locations of SNPs in increasing order. **Please note: all SNP positions should be encoded as in kb rather than bp.** The locations can be separated by "\n"s (as in the example), blanks, or tables.

For a file of haplotypes / genotypes, three elements should appear in the first line: (1) number of genotypes / haplotypes, (2) number of SNPs, (3) phased data (1) or unphased data (2). The second element should be the same as the number of SNPs in the input file of SNP locations. Following the head line, the file gives haplotypes / genotypes. Each haplotype / genotype is in a modified FASTA format, and the comment line of it can be an arbitrary string. **The sequence line cannot be separated into several parts and occupy several lines.** It must be in a single line; no matter how long the single line is, the program will read it without problem. The polymorphisms of a SNP are denoted as `0` and `1` in haplotypes, and `0`, `1` (homozygotes) and `2` (heterozygote) in genotypes. Missing values are denoted as `?`. For gaps or insertions, users need to convert it into `0`/`1` or `0`/ `1`/`2` beforehand.

A simple example for the file of SNP locations is as follows:

```
9 20.000 L
10.401
11.509
12.570
12.899
13.081
15.301
15.913
16.567
17.352
```

A simple example for the file of haplotypes / genotypes is as follows:

```
3 9 2
>geno1
222112010
>geno2
011120112
>geno3
011002121
```

HotspotFisher also requires a lookup table file that includes two-locus log-likelihoods for all possible combinations in a given number of haplotypes. For example, if the data contains 90 haploid / phased data, a table for 90 haplotypes is required. If the data contains 90 diploid / unphased data, a table for 180 haplotypes is required. This table can be generated using either ms (Hudson 2002) or LDhat 2.0 (McVean et al. 2004). However, to save time, we strongly suggest using a pre-computed one. Several lookup tables for different numbers of haplotypes are available in http://www.stats.ox.ac.uk/~mcvean/LDhat/instructions.html, with the same format as HotspotFisher uses, and users may use *lkgen* in LDhat 2.0 package to generate a proper lookup table from a table for larger number of haplotypes.

3. Using the program

To run HotspotFisher from the command line, type

```
% ./HotspotFisher lookupTable inLocs inData outRes [-F MAF]
[-P effPopuSize] [-W effWinLen] [-T LLRThrd]
```

e. g.

```
HotspotFisher lk_n180_t0.001 locs geno res
```

e.g.

```
HotspotFisher lk_n180_t0.001 locs geno res –P 20000 –F 0.15
```

The 4 parameters without brackets must be set in turn. They are: (1) the file name of the lookup table, (2) the file name of SNP locations, (3) the file name of haplotypes / genotypes. (4) the file name of results (an arbitrary string).

The parameters in brackets are optional, and they may appear in arbitrary turn after the 4 parameters above. These parameters are:

-F: minor allele frequency (MAF) cutoff: only SNPs with MAF bigger than this threshold is used to do hotspot inference. Its default value is 0.05.

-P: Effective population size (*Ne*). Its default value is 10,000. We recommend using 10,000 for non-African population, and 20,000 for African population.

-W: the length of effective neighborhood windows (*N*). Its default value is 7. Please refer to my manuscript for a detailed description.

-T: the LLR threshold that controls the false-positive rate. Its default value is 26, which will give a positive rate of 0.4 per Mb for *N* = 7.

The speed of the program is mainly determined by number of SNPs in the region, so we recommend that users divide a long region into some ~200 kb segments. What also affects the speed is the level of missing data, since HotspotFisher have to calculate the two-locus likelihoods for the missing data before inferring hotspots.

4. Output files

A single file recording the results is generated while the program is done. The first line shows the information about the background. The first element is set as `0`, and the last element is set as `0.000`, with no particular meanings. The second and third elements are the start and end sites of this region, which is equal to the positions of the first and last SNPs. The fourth element is the background recombination rate. Each of the lines following shows the information of a hotspot in turns of (1) index, (2) start position, (3) end position, (4) population-scaled recombination rate, (5) LLR. The index of a hotspot is the order that this hotspot is detected. The hotspot detected first has an index `1`, the hotspot detected second has an index `2`, and so on.

Here is an example:

```
0  0.001       205.313     0.120    0.000
1  108.801     110.401     143.216  76.987
2  69.401      71.201      13.517   75.305
3  199.201     201.201     4.250    58.428
4  166.801     169.201     5.876    41.514
5  189.601     192.001     7.071    36.440
6  42.401      44.601      3.699    30.091
```

In this example, 6 hotspots are detected in the region (0.001 kb ~ 205.313 kb), and the background rate of the region is 0.120 per kb per generation. The hotspot detected first extends from 108.801 kb to 110.401 kb, with LLR 76.987, and its recombination rate is 143.216 per kb per generation. The hotspot detected second extends from

69.401 kb to 71.201 kb, with LLR 75.305, and its recombination rate is 13.527 per kb per generation.

5. Bugs

This program is generally stable and functions appropriately if all the instructions are followed. If you find any bugs, errors, or have suggestions, please report to jun-li00@mails.tsinghua.edu.cn. Thank you in advance!

6. Acknowledgements

We use some codes in LDhat 2.0 for inferring two-locus likelihoods of genotypes from those of haplotypes, and inferring two-locus likelihoods with missing data. Many thanks to Gil McVean.

References

Hudson RR (2002) Generating samples under a Wright-Fisher neutral model of genetic variation. Bioinformatics 18:337-338

McVean GA, Myers SR, Hunt S, Deloukas P, Bentley DR, Donnelly P (2004) The fine-scale structure of recombination rate variation in the human genome. Science 304:581-584