FISHR2 Documentation

Matthew C Keller & Douglas W. Bjelland

Institute for Behavioral Genetics
1480 30th Street
Boulder, Colorado, 80303

February 20, 2017

1 Introduction

FISHR2 is written in C++ and is freely available for download at `https://github.com/matthew-c-keller/FISHR2.git`. FISHR2 attempts to detect segments of the chromosome shared identical by descent (IBD) between all pairs of individuals in a sample who are measured on phased genomewide SNP data. FISHR2 utilizes a modified version of GERMLINE (Gusev et al, 2009) as an initial screen to quickly detect candidate IBD segments (Figure 1). By default, FISHR2 uses the `h_extend` method in GERMLINE, which incorporates information on phased mismatches, to detect candidate segments between two individuals or within individuals (runs of homozygosity). FISHR2 then further refines the candidate segments as follows. First, because two long IBD calls separated by a short distance may actually be a single contiguous IBD segment that was artificially broken apart in GERMLINE due to phase or SNP call errors, FISHR2 stitches together segments separated by a user-defined number of SNPs (`-gap`). Next, FISHR2 finds the locations of implied errors (IEs) likely SNP call or phase errors - for all called segments. To do this, FISHR2 finds the longest exact match between either of the two phased haplotypes of the first person and either of the two phased haplotypes of the second person (a total of four possible combinations), starting at the first SNP of the called segment. An IE occurs at the mismatching SNP after the exact haplotypic match ends. FISHR2 then finds the next longest exact match between any of the four possible combinations of phased haplotypes, starting from the SNP following the previous IE, and extends until the next mismatching SNP is encountered. This process continues until the end of the called segment. The locations of each of these IEs along a candidate segment is stored and used as described below to better find the endpoints of IBD segments.

IEs represent locations along a candidate segment that are potentially inconsistent with IBD inheritance. Some IEs are expected by chance due to SNP and phase errors even in truly IBD segments. However, too many IEs within a particular region are a likely signal that the segment is not IBD in that area and that the segment should be truncated (if near an endpoint of the segment) or split into two (if in the middle of the segment). To determine such called segment endpoints, FISHR2 calculates a moving average (MA) of IEs centered at each SNP within a user-defined window (using the `window` flag) of SNPs. FISHR2 then starts at the center of the called IBD segment and moves towards each endpoint until it reaches the first SNP with a MA value greater than the user-defined maximum (`emp-ma-threshold`). These points signal the endpoints of a called segment (Figures 2 and 3). One source of error in IBD detection could be the erroneous merging of separate distinct IBD segments. Therefore, in addition to trimming the segment ends, this process can split a GERMLINE candidate segment into two or more shorter segments, depending on the distribution of the MA values. Moreover, if the flag `count-gap-errors` is set to TRUE, as it is by default, segments that had been stitched together from the first step can be broken up again at this stage if enough IEs are clustered near the gap. Because segments that are too short, in terms of either number of SNPs or cM distance, are increasingly likely to be false positives, FISHR2 drops segments shorter than user-defined thresholds of both SNP and cM length (using the `min-snp` and `min-cm` flags, respectively). Finally, FISHR2 calculates the total proportion of SNPs that are IEs (PIE) within each remaining segment. Too many IEs scattered across the entire length of a segment are a signal that the whole segment is unlikely to be IBD. Thus, if the PIE of a segment is greater than the value supplied in the `emp-pie-threshold` argument, the segment is dropped (Figure 4).

For a given cM threshold (defined by `min-cm`), the performance of FISHR2 depends strongly upon user-supplied values of `emp-pie-threshold` and, especially, `emp-ma-threshold`. High values of these parameters lead to more false positive IBD segments and/or segments whose endpoints are over-extended. Low values of these parameters lead to more false negative IBD segments and/or segments whose endpoints are under-extended. These parameters are therefore crucial in determining the trade-off in these two statistics, and will depend on the user's needs in a particular context. We supply a utility, `parameter_finder`, to help users decide on values of these parameters that are optimal for the given context.

FISHR2 assumes that you have split your genome such that each file of SNP data spans at most one chromosome. Users may also split chromosomes (e.g., at the centromere), but files spanning multiple chromosomes will not work in FISHR2.

2 Changes from original FISHR program

FISHR2 is the second version of the FISHR program. The main changes in FISHR2 over FISHR are:

- The original version (FISHR) required that GERMLINE2 (a modified version of the GERMLINE software) be run first and then FISHR be called on the segments detected by GERMLINE2. This is no longer the case. FISHR2 can now be used directly on the phased SNP data, and outputs shared haplotypes in the same format as before. Thus, GERMLINE2 is run internally, which simplifies using FISHR2. There is an option that allows users to write out the intermediate GERMLINE2 output if they wish.

- FISHR2 can now use phased files directly from SHAPEIT output (*gen, *hap, and *sample), as well as phased *ped files as before. This greatly simplifies using FISHR2 because phased *ped file is not a common format, being used only by GERMLINE/GERMLINE2 and FISHR/FISHR2.

- FISHR2 can now detect shared haplotypes that are IBD2 and IBD4 (i.e., two or four overlapping segments shared IBD between two individuals). Previously, FISHR could only detect instances of IBD1; IBD2 or IBD4 would be called as IBD1. This was not much of a problem in unrelated samples (the probability of IBD2 is roughly the probability of IBD1 squared), but was a considerable limitation in datasets including siblings.

3 FISHR2 Input Files

There are two options of sets of files that can be used, phased PED files or SHAPEIT phased files:

Phased PED file input

`-ped-file` [file path] - The location (path) to a single phased PED file. This is a standard plink *.ped file with one exception: it MUST be phased. Phased PED files look like normal PED files (with genotypes at each SNP separated by whitespace or tab), but in phased PED files the alleles to the left of the whitespace (or tab) are all predicted to be on the same haplotype and similarly for those to the right of the whitespace (or tab). We have provided utility programs (e.g., gap) that can take a chromosome that was phased using SHAPEIT2 or BEAGLE and output a phased PED file. The paternal ID, maternal ID, sex, and disease status variables do not affect the FISHR2 output. For more information on phased PED files, see `http://www.cs.columbia.edu/~gusev/germline/`

The columns of a phased PED file correspond to: Family ID (FID), Individual ID (IID), Paternal ID, Maternal ID, Sex, Disease Status, allele1.SNP1 allele2.SNP1 allele1.SNP2 allele2.SNP2 allele1.SNP3 allele2.SNP3. Here, allele1.SNP1, allele1.SNP2, and allele1.SNP3 are all predicted to be in phase (exist on the same haplotype), and similarly for allele2.SNP1, allele2.SNP2, and allele2.SNP3.

`-mapfile` [file path] The location (path) to a single MAP file. This is a standard plink *.map file. The genetic map positions (cM) in the third column must exist and correspond as closely to the dataset being used to provide as accurate of estimates as possible. The columns correspond to: Chromosome number, SNP name, centimorgan (cM) distance, basepair position.

SHAPEIT file input

-mapfile [file path] The location (path) to a single MAP file. This is exactly the same map file described above.

-samplefile [file path] - The location (path) to a single sample file produced by SHAPEIT.

-hapsfile [file path] - The location (path) to a single haps file produced by SHAPEIT.

4 FISHR2 Parameters

-bits [integer] This is the length in SNPs of the initial candidate segment discovered by the internal GERMLINE2 algorithm before any GERMLINE2 extension occurs. The chromosome is broken up into non-overlapping segments (”slices”) with a length in SNPs of this parameter. The initial segment discovery then finds exact matches between individuals of these non-overlapping slices. Smaller values (< 30) will result in more potential segments being discovered, but a higher number of false positive IBD segments being fed to FISRH2 as well as a much longer runtime. Longer values (> 120) will run much more quickly, but at the cost of missing potential IBD segments.

-err_hom [integer] The maximum number of mismatching homozygous markers for a slice to still be considered part of a match.

-err_het [integer] The maximum number of mismatching heterozygous markers for a slice to still be considered part of a match

-min_cm_initial [numeric] The minimum length, in cM, the candidate segments discovered by GERMLINE2 need to be. So that FISHR2 can merge shorter segments that were incorrectly split by GERMLINE2, we recommend that this minimum cM length is less than the final desired cM length output by FISHR2. For example, if you want to look at IBD segments ¿ 3 cM, then set this parameter to be, say, 1.5, and then set the minimum in FISHR2 to be 3 cM.

-homoz A flag. This is option has no parameters and is either included or not. If it is included, FISHR2 will attempt to detect runs of homozygosity within an individual in addition to detecting IBD segments between individuals.

-w_extend - A flag. If used alone (not in conjunction with h_extend), this extends the slice until the first opposite homozygote occurs in each direction. If used in conjunction with h_extend (as we recommend!), this option extends the segment until the number of phased mismatches exceeds some number, as defined by -err_het.

-h_extend - A flag. The option -h_extend should always be used in conjunction with -w_extend (so either use -w_extend alone, which uses opposite homozygotes to determine segment endpoints, which tend to be longer, or use -h_extend and -w_extend together, in which case phase information is also used to determine segment endpoints). FISHR2 has been optimized to run with using -h_extend and -w_extend, but it's fine to use -w_extend alone if the user wishes to have more a higher false positive and lower false negative rate of initial candidate segments. FISHR2 will work fine for both, but tends to be more efficient and produce slightly more accurate results when both are used together.

-min_cm_final [numeric] The minimum length, in cM, of the final segment calls made by FISHR2; segments smaller than this (either candidate segments that are initially too small from the GERMLINE2, or segments that become too small after trimming by FISHR2) will be dropped.

-min_snp [numeric] The minimum length, in SNPs, of the final segment calls made by FISHR2; segments smaller than this (either candidate segments that are initially too small from the GERMLINE2, or segments that become too small after trimming by FISHR2) will be dropped.

-window [integer] The length, in SNPs, of the moving window used to calculate the moving average of IEs by FISHR2.

-gap [integer] The maximum distance, in SNPs, between two called IBD candidate segments from GERMLINE2 that are combined into one segment by FISHR2. Note that, if evidence so indicates from a high enough moving average, these segments can be broken apart again by FISHR2, so long as `-count.gap.errors` is set to TRUE.

`-count.gap.errors` [TRUE/FALSE] - This parameter is either TRUE or FALSE. If TRUE (the default), FISHR2 will count the implied errors within the gap when calculating the `-emp_ma_threshold` and the `-emp_pie_threshold`. If this is FALSE (not recommended), then once joined, the MA step will not break up SHs at the gap junctures.

`-emp-ma-threshold` [numeric] The maximum value for the proportion of implied errors within a window of SNPs before a segment is terminated. The termination occurs at the middlemost SNP within the window. A lower value (e.g., .03) will result in shorter segments being discovered, with a higher likelihood that the ends are truly IBD but often under-extended, while a higher value (e.g., .08) will result in longer segments where the endpoints may be over-extended. The utility parameter_finder can help users determine the optimal value of this parameter.

`-emp-pie-threshold` [numeric] The maximum proportion of implied errors that are allowed in a final called segment. A large value (e.g., .04) will allow more errors, and thus, potentially result in segments that are not truly IBD but a higher sensitivity. A smaller value (e.g., .01) will be more conservative, allowing fewer errors and resulting in segments that are more likely to be IBD but at the cost of more false negatives. The utility parameter_finder can help users determine the optimal value of this parameter.

`-output_type` [character] - This determines which one of several different output files are generated. *finalOutput* is the default and the one most users should use. Other options are *finalErrorsOutput* (as *finalOutput*, except adds a final column that is PIE of each SH), *Reduced* (like *finalOutput* except creates smaller files by omitting 3 columns), *FullPlusDropped* (like *finalErrorsOutput* but also includes the SH's that were dropped and the reason for the drop), *Error1* (outputs error information for all the SHs that existed in the original GERMLINE2 file after consolidation, and includes three additional columns: start/stop place (in SNPs) for SH; location of each error, separated by "/"; and the PIE for each SH), and *Error2* (outputs error information for the final SHs; the number of rows for this file will therefore be the same as that for *finalOutput*).

`-log-file` [path] - File location where the log information is written.

`-ibd2` [numeric]- *OPTIONAL* - This is the minimum cM threshold for IBD2 and IBD4 segments; typically, we want this about the same length or a bit shorter than the `-min_cm_final` argument. The moving average for detecting IBD2 and IBD4 segments is set at the value of `-emp-ma-threshold`.

`-germline_output` - *OPTIONAL* - name of the folder where GERMLINE2 (intermediate) output goes; if this parameter is not used, then no GERMLINE2 output will be created

5 Procedure

Running on phased PED formatted files

```
FISHR2 -mapfile Test.SI.map -pedfile Test.SI.sample -bits 60 -err_hom
0 -err_het 0 -min_cm_initial 1.5 -homoz -w_extend -h_extend -min_cm_final
3 -min_snp 64 -window 50 -gap 5 -output-type finalOutput -count.gap.errors
TRUE -emp-pie-threshold 0.015 -emp-ma-threshold 0.045 -log-file
SI.Test.8k | gzip > Test.8k.PEDFormat.FISHR2.gz
```

Running on SHAPEIT formatted files

```
FISHR2 -mapfile Test.SI.map -samplefile Test.SI.sample -hapsfile
Test.SI.haps -bits 60 -err_hom 0 -err_het 0 -min_cm_initial 1.5 -homoz
-w_extend -h_extend -min_cm_final 3 -min_snp 64 -window 50 -gap 5
-output-type finalOutput -count.gap.errors TRUE -emp-pie-threshold
0.015 -emp-ma-threshold 0.045 -log-file SI.Test.8k | gzip > Test.8k.SIFormat.FISHR2.gz
```

Running on SHAPEIT formatted files, and detecting IBD2 and IBD4 segments

```
FISHR2 -mapfile Test.SI.map -samplefile Test.SI.sample -hapsfile
Test.SI.haps -bits 60 -err_hom 0 -err_het 0 -min_cm_initial 1.5 -homoz
-w_extend -h_extend -min_cm_final 3 -min_snp 64 -window 50 -gap 5
-output-type finalOutput -count.gap.errors TRUE -emp-pie-threshold
0.015 -emp-ma-threshold 0.045 -ibd2 2 -log-file SI.Test.8k | gzip >
Test.8k.SIFormat.FISHR2.gz
```

Using the data supplied with FISHR2 in the src folder, the first two above procedures will identify 392,994 IBD segments > 3 cM in length and that passed the PIE threshold of .015 and MA threshold of .045 suggested above. GERMLINE2 originally detected 3,709,454 segments > 1.5 cM (the min_cm_initial parameter), and these were passed on to FISHR. 1,609,436 of these were dropped straight off (after any merging 80,746 separated by fewer than 5 SNPs), then 1,626,234 were dropped that were originally > 3cM but that trimming them based on the Moving Average of IEs made them < 3cM, and finally 44 (only) were removed because they had an overall proportion of IEs (PIE) > .015. Note that the ”Total time” output by the algorithm in the log file is that used by the FISHR part of the program and does not consider the initial and longer GERMLINE part of the algorithm.

The final procedure output is almost the same as the first two, except that you will have discovered 398,519 IBD segments, 5,525 more than the first two runs. These additional 5,525 segments are estimated IBD2 and IBD4 segments.

6 Output Files

Note that FISHR2 writes to standard out, so needs to be piped to (e.g., gzip) and written to file. Using the -finalOutput option, FISHR2 writes out a gzipped file with the following columns:
Individual 1
Individual 2
Start position in bp of the final segment
End position in bp of the final segment
Length in SNPs of the final segment
Length in cM of the final segment

7 parameter_finder utility for finding optimal parameters to use in FISHR2

A utility program, parameter_finder, is supplied alongside FISHR2 (located in the utilities folder) to help users choose values for `emp_pie_threshold` and `emp_ma_threshold`, which influence the tradeoff between false positives and false negatives the most strongly (see Figure 5). To do this parameter_finder provides distributions of PIE and MA from segments in the users data that are truly IBD (defined as the middlemost 50% of all stretches of the genome > 8 cM that contained no opposite homozygotes between pairs of individuals) and similar distributions of PIE and MA from non-IBD segments (defined as segments between random pairs of individuals with the same start and endpoints as the truly IBD segments). Users can then compare the IBD and non-IBD distributions of PIE and MA to choose PIE and MA thresholds that produce a desired sensitivity or specificity, depending on the users priorities.

Users must first run GERMLINE2 to find long candidate segments. Unfortunately, GERMLINE2 currently only runs on phased PED file formats.

Here is an example of running GERMLINE2 (using only opposite homozygotes to determine them so as not to falsely remove long ones based on phased errors) to find long segments, the middlemost portion of which are almost certainly IBD:

```
GERMLINE2 -pedfile Test.SI.ped -mapfile Test.SI.map -outfile Test.SI
-bin_out -err_hom 0 -err_het 0 -reduced -bits 120 -min_m 8 -w_extend
```

`-pedfile` - a PHASED pedfile.

`-mapfile` - a map file with cM distances in 3rd column

`-bin_out` - a compressed output necessary for being read by parameter_finder

`-err_hom` 0 - allow 0 mismatching homozygous markers

`-err_het` 0 - allow 0 mismatching heterozygous markers

`-reduced` - a flag telling GERMLINE2 to reduce the columns of the output; necessary for being read by parameter_finder and FISHR

`-bits` 120 - the number of SNPs in each fixed window that GERMLINE uses for initial matches

`-min_m` - minimum length in cM for match to be output; here 8

`-w_extend` extend match beyond the "bits" window until the first opposite homozygote (OH) occurs. Note that we do NOT use the `-h_extend` option when running GERMLINE to find optimal parameters. This is because using phase information will bias the detected SHs to have fewer phase errors than randomly chosen truly IBD segments, and will make our sensitivity and specificity values based on the chosen thresholds below appear better than they really would be in real circumstances. Although it's true that OHs also cause IEs, they are a large minority of IEs, most of which are caused by phase and SNP errors.

Example of running parameter_finder:

```
parameter_finder -bmatch Test.SI.bmatch -bsid Test.SI.bsid -bmid Test.SI.bmid
-ped-file Test.SI.ped -window 50 -cut-value 0.5 -reduced 500 8 -output-type
Error1 -log-file Test.SI.paramfind | gzip > Test.SI.PF.gz
```

`-bmatch` - the binary output (from -bin_out) of GERMLINE; has one SH per row

`-bsid` - the subject IDs output from GERMLINE

`-bmid` - the marker IDs output from GERMLINE

`-ped-file` - the path to the phased data; should be identical to the input -pedfile used in GERMLINE

`-window` 50 - says to use a 50 SNP window for figuring moving average of IEs (MA)

`-cut-value` .5 - says to trim the SH to the middlemost 50%: 25% from the left and 25% from the right are trimmed. This ensure that the remaining segment is IBD

`-reduced` 500 8 - recommended paramers in real data; only use initial SHs that are at least 500 SNPs long and > 8cM. The outputed middlemost segments will typically be < 8cM (e.g., 4+ cM)

`-output.type` Error1 - this is the typical output format to be requested.

| gzip > this pipes the standard out to gzip so the final output is a gzipped file.

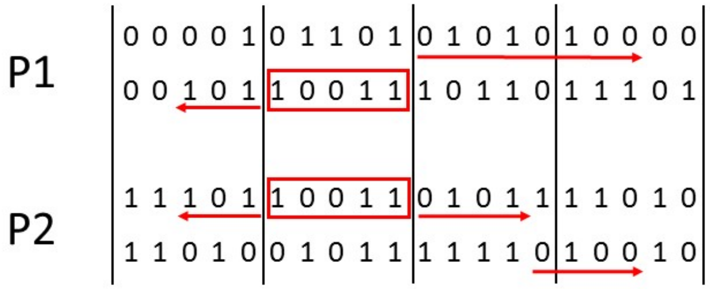

Figure 1: This figure displays the internal GERMLINE2 process of finding a potential segment and the extension process. The chromosomal segment is divided into smaller segments with a length defined by the `-bits` argument (in this case `-bits` 5) as denoted by the vertical lines. The perfect match in the smaller segments is denoted by the square around the variants. Then, the segment is extended until an opposite homozygous SNP is detected (if using only the `-w_extend` flag in GERMLINE2) or until enough phased mismatches occur (if using both the `-w_extend` and `-h_extend` flags).

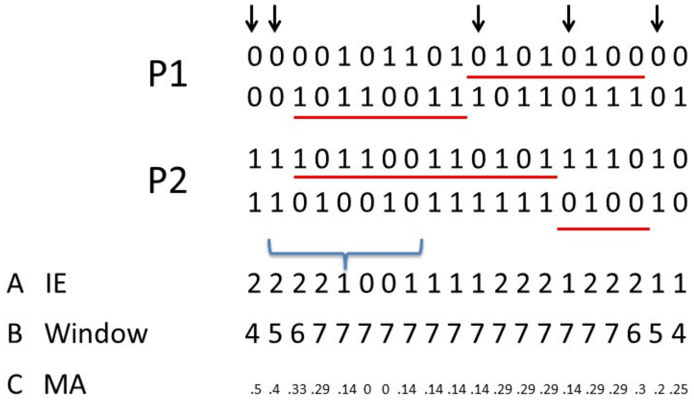

Figure 2: The moving average of IEs (MA) is calculated for each SNP within and around the called segment. In this example the `-window` value is 7, each SNP will use that SNP, as well as the three SNPs on either side of it, to calculate the MA value for that particular SNP.

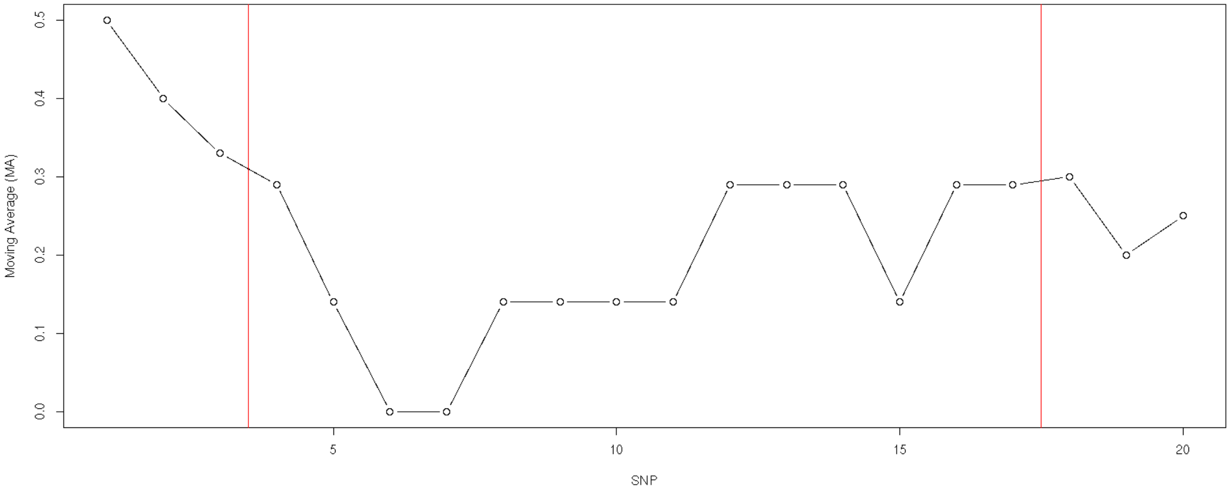

Figure 3: Once all the MA values are calculated, FISHR2 then starts in the center of the segment and move outward until a value is reached that is greater than the `-emp_ma_threshold` value. The point at which the MA for a SNP is above this value will determine the end points of the called segment. In this example the `-emp_ma_threshold` is 0.05 and the IBD segment will be called with the endpoints at SNP 4 and 17.

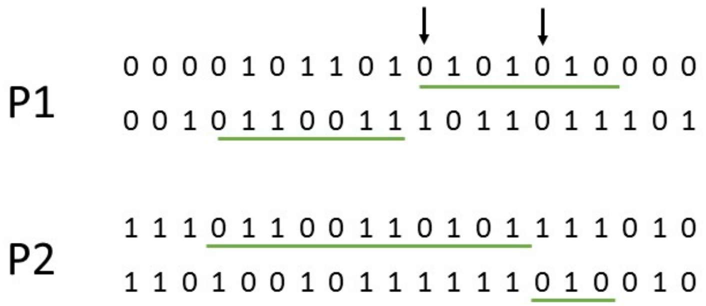

Figure 4: Once the endpoints are determined using the `-emp_ma_threshold`, the total number of IEs is calculated. If this value is greater than that provided by the `-emp_pie_threshold`, the entire segment will be removed.

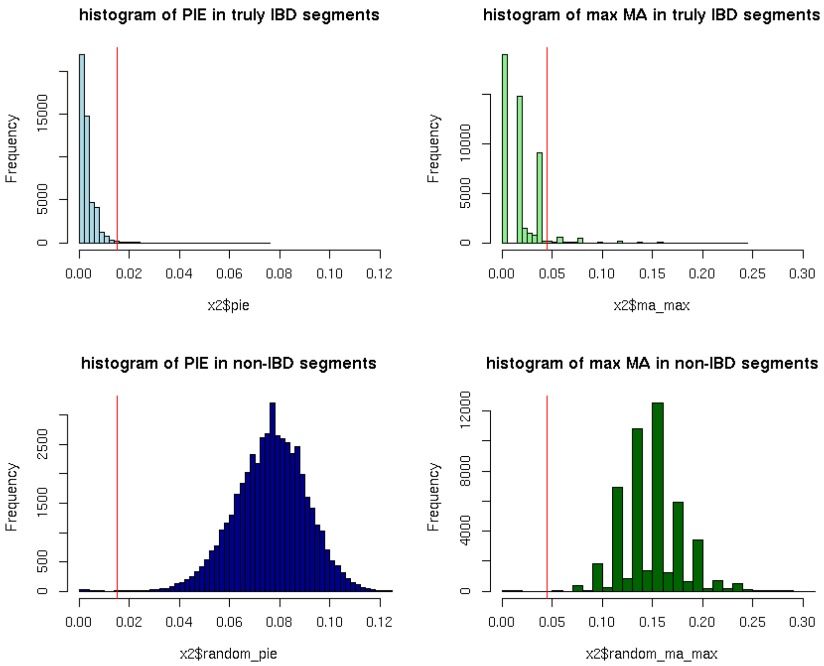

Figure 5: The top row shows the distributions of PIE and MA, respectively, in the middle 50% of very long segments, which are almost certainly IBD. Similarly, the bottom row shows the distributions of PIE and MA, respectively, in the middle 50% of segments at the same endpoints as the long discovered segments but which are between random pairs of individuals, and therefore almost certainly not IBD. The user can use these distributions to determine optimal MA and PIE thresholds, according to their needs for limiting either false positive or false negative IBD calls. Here, a `-emp_pie_threshold` of .015 is expected to lead to few false positives (the upper left histogram) and few false negative (the lower left histogram). Moreover, a `-emp_ma_threshold` of .045 is expected to lead to low amount of under-extension (the upper right histogram) and over-extension (the lower right histogram). Note that this approach does not account for the increasing uncertainty that occurs at SH endpoints, where IEs begin to accumulate. In real data and for short IBD segments, the IEs ocurring at endpoints make up a greater and greater share of the total IBD segment length and increase the false negative and false positive rate.