FISHR2 Documentation

Matthew C Keller & Douglas W. Bjelland

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

February 28, 2018

1 Overview

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), which we call GERMLINE2, as an initial screen to quickly detect candidate IBD segments (Figure 1). By default, FISHR2 uses the `-h_extend` method in GERMLINE2, 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 GERMLINE2 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 are stored and used to better find the endpoints of IBD segments, as described below.

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 the middlemost SNP within a window of user-defined SNP length (using the `-window` flag). FISHR2 then starts at the center of the called IBD segment and moves in both directions, towards each endpoint, until it reaches the first SNP with a MA value greater than the user-defined maximum (`-emp-ma-threshold`). These points define the endpoints of a called segment (Figures 2 and 3). Additionally, GERMLINE2 could have erroneous merged two or more distinct IBD segments into one. Therefore, in addition to trimming the segment ends, this MA trimming process can split a segment into two or more shorter segments if the MA values surpass those supplied in `-emp-ma-threshold`. Moreover, if the flag `-count-gap-errors` is set to TRUE, as it is by default, two candidate segments that had been stitched together from FISHR2's 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 parameter values of these arguments

lead to more false positive IBD segments and/or segments whose endpoints are over-extended. Low parameter values of these arguments 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` along with FISHR2 (located in the `utilties` folder), 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. Generally, this is preferable anyway, because it allows IBD detection to be done in parallel.

We HIGHLY recommend that users first work through the example R script provided in the FISHR2 download before attempting to use 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 (*hap and *sample), as well as phased PED files (*ped) as before. This greatly simplifies using FISHR2 because the 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 Arguments

3.1 File Input Options

There are two sets of file types that can be used as input to FISHR2, 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 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. Columns 3-6 (paternal ID, maternal ID, sex, and disease status) are ignored and do not affect the FISHR2 output. For more information on phased PED files, see `http://www.cs.columbia.edu/~gusev/germline/`

`-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.

3.2 FISHR2 required arguments

`-bits` [integer] This is the length in SNPs of the initial candidate segment discovered by the internal GERMLINE2 algorithm before any GERMLINE2 extension (via `-w_extend` and `-h_extend`) occurs. The chromosome is broken up into non-overlapping segments ("slices") with a length in SNPs of this argument. 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 and a longer runtime. Longer values (> 120) will run more quickly, but at the cost of missing true 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. Alternatively, if the user is unconcerned with (the fairly low proportion) of segments that will be stitched back together by FISHR2, the value of this argument can be set to be just slightly smaller than the value specified in the `-min_cm_final` argument, which has the advantage of a large speedup and much reduced RAM footprint in FISHR2. (Currently, FISHR2 will crash if `-min_cm_initial` is equal to or greater than `-min_cm_final`).

`-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 generally 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 should not be used alone (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 is accurate for either procedure, 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` [integer] 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 (MA) 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 segments 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 argument.

`-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 argument.

`-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 segment), *Reduced* (like *finalOutput* except

creates smaller files by omitting 3 columns), *FullPlusDropped* (like *finalErrorsOutput* but also includes the segment's that were dropped and the reason for the drop), *Error1* (outputs error information for all the segments that existed in the original GERMLINE2 file after consolidation, and includes three additional columns: start/stop place (in SNPs) for segment; location of each error, separated by "/"; and the PIE for each segment), and *Error2* (outputs error information for the final segments; the number of rows for this file will therefore be the same as that for *finalOutput*).

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

3.3 FISHR2 Optional Arguments

`-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.

`-ibd2` [numeric] 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 parameter defined in the `-min_cm_final` argument. The moving average for detecting IBD2 and IBD4 segments is set at the value of `-emp-ma-threshold`. UPDATE: We no longer recommend that users use the `-homoz` argument and the `-ibd2` arguments simultaneously; they interact in weird and unforeseen ways. `-ibd2` works fine if you omit the `-homoz` flag, which means that self-matching IBD segments (runs of homozygosity) will be omitted, and IBD4 segments will (necessarily) no longer be called.

`-germline_output` [folder path] This is the location of the folder where GERMLINE2 (intermediate) output goes; if this argument is not used, then no GERMLINE2 output will be created

4 Examples

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

Example output 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 argument), and these were passed on to FISHR. 1,609,436 of these were dropped straight off (after merging 80,746 segments 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. Given the values of -emp-pie-threshold and -emp-ma-threshold used, it is normal for very few segments to be dropped due to PIE because most have already been removed based on endpoint truncation; higher values of -emp-ma-threshold

and lower values of `-emp-pie-threshold` would cause many more to be removed due to PIE, but we have found performance to be optimized using the argument values supplied above. Note that the "Total time" output in the log file is currently (we will change this) that used by the FISHR subroutine and does not consider the initial and longer GERMLINE subroutine of the algorithm.

The output of the final run (when using `-ibd2`) 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. This number is expected to be low in unrelated samples, but will be substantially higher in samples that include siblings.

5 FISHR2 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.

In addition, if using the `-ibd2` argument, FISHR2 will output a final column denoting whether the segment is IBD1, IBD2, or IBD4. IBD2 segments will be placed directly underneath the (by definition) larger, overlapping IBD1 segments shared between the same pairs of individuals. IBD4 segments will always begin with a single row of IBD1 followed by one row of IBD2 and, finally, two rows of IBD4 segments.

6 Using the parameter_finder Utility for Finding Optimal Parameters to Use in FISHR2

Most users will be fine using the values for `-emp_pie_threshold` (=.015) and `-emp_ma_threshold` (=.045) that we recommend. For users that wish to explore alternative (and potentially better) thresholds for their specific purpose, a utility program, `parameter_finder`, is supplied alongside FISHR2 (located in the `utilities` folder), which can be run before running FISHR2 to help users choose values for `-emp_pie_threshold` and `-emp_ma_threshold`, which influence the tradeoff between false positives and false negatives (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 portion 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. Users should only use `-w_extend` option only and NOT in conjunction with the `-h_extend` option when doing this. The `-w_extend` option only uses opposite homozygotes to define endpoints, which leads to a very low rate of false negative calls, such that few truly IBD segments will be missed. At a very long length (e.g., 8+ cM), false positive IBD segments become extremely unlikely, although `-w_extend` WILL tend to over-extend segments. We therefore use the middlemost (e.g., 50%) of segments to ensure that (nearly) all segments we think are IBD truly are IBD when calculating PIE and MA distributions.

Although the use of the `-w_extend` option means that the PED files don't need to be phased for GERMLINE2 to detect IBD segments, `parameter_finder` requires phased data to determine distributions of PIE and MA in the truly IBD segments and in the non-IBD segments, and therefore phased PED file format is currently required for use of `parameter_finder`. Although this is a nuisance for users wanting to use SHAPEIT formatted data in FISHR2, `parameter_finder` probably only needs to be run on a subset of the data (e.g., one or a few chromosomes, and potentially only a random subset of the sample if it is large enough) to determine optimal thresholds.

6.1 Step 1: Running GERMLINE2

We use only opposite homozygotes to determine segment endpoints in this procedure so as not to falsely remove truly IBD segments based on phased errors. However, this tends to over-extend segments. To ensure that our MA and PIE distributions are from truly IBD sections of these segments, we then use only the middlemost portion of them, which are almost certainly IBD:

6.1.1 GERMLINE2 arguments:

`-pedfile` [file path] A PHASED pedfile.

`-mapfile` [file path] A map file with cM distances in 3rd column.

`-bin_out` [flag] A compressed output necessary for being read by `parameter_finder`.

`-err_hom` [integer] Number of mismatching homozygous markers allowed in matching slices.

`-err_het` [integer] Number of mismatching heterozygous markers allowed in matching slices.

`-reduced` [flag] A flag telling GERMLINE2 to reduce the columns of the output; necessary for being read by `parameter_finder`.

`-bits` [integer] The number of SNPs in each fixed window that GERMLINE uses for initial matches.

`-min_m` [numeric] Minimum length in cM for match to be output.

`-w_extend` [flag] Extends matching slices 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 segments 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.

6.2 Step 2: Running `parameter_finder`

`parameter_finder` takes the very long segments output by GERMLINE2 and finds the distributions of MA and PIE within the user-defined middlemost portion of them - these are considered good estimates of MA and PIE in truly IBD segments. `parameter_finder` then finds the MA and PIE distributions at the same start and endpoints among random pairs of individuals. These are considered good estimates of MA and PIE distributions in non-IBD segments. These distributions can be compared to find optimal values of `-emp_pie_threshold` and `-emp_ma_threshold` (see Figure 5).

6.2.1 `parameter_finder` Arguments:

`-bmatch` [file path] The binary output (from `-bin_out`) of GERMLINE2; has one segment per row.

`-bsid` [file path] The subject IDs output from GERMLINE2.

`-bmid` [file path] The marker IDs output from GERMLINE2.

`-ped-file` [file path] The path to the phased data; should be identical to the input `-pedfile` used in GERMLINE2, i.e., a phased PED file.

`-window` [integer] Size of the SNP window for figuring moving average of IEs (MA).

`-cut-value` [numeric] The middlemost portion of segments to use in reporting MA and PIE values. For example, .5 says to trim the segments to the middlemost 50%: 25% from the left and 25% from the right are trimmed. This ensure that the remaining segment is almost certainly IBD.

`-reduced` [numeric] [integer] The minimum length in SNPs and cM for `parameter_finder` to consider the segment truly IBD. The recommended parameters in whole-genome SNP data (of 1M SNPs) are 500 and 8, which says only use initial segments that are at least 500 SNPs long and > 8cM. The outputed middlemost segments will typically be < 8cM (e.g., 4+ cM if `-cut-value` is .5)

`-output.type` [character] *Error*1 is the typical output format to be requested.

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

6.3 Example usage of GERMLINE2 and `parameter_finder`

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`

`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

7 Performance of FISHR2

FISHR2 is fast enough to be used on very large SNP datasets (e.g., > 60 000 individuals), running two to three times slower than the original GERMLINE program but running over a thousand times faster than alternative programs, such as rIBD and HaploScore, at large sample sizes (see Table 1).

One practical downside of FISHR2 is that it requires much more RAM than GERMLINE because all candidate segments from GERMLINE2 need to be held in RAM to be sorted in order to be (potentially) stitched together. We developed a version of the original FISHR program that used a negligible amount of RAM at the cost of failing to stitch together called segments that are erroneously split, but this has yet to be implemented in FISHR2. In the meantime, if limited RAM is an issue for using FISHR2 (on very large datasets or nodes with limited RAM), users can specify that `-min_cm_initial` is only slightly less than `-min_cm_final` (e.g., 2.9 and 3.0

respectively). This will mean that many fewer candidate segments passed from GERMLINE2 need to be sorted by FISHR2. (Note that FISHR2 will crash if `-min_cm_initial` is greater than or equal to `-min_cm_final`). If this still does not solve your RAM limitation, we recommend that you use FISHR_Low_Ram, which can be downloaded with the original version of FISHR on gihub. This does not have the functionality of FISHR2 (e.g., cannot use SHAPEIT formatted file, cannot call IBD2, and does not stitch IBD segments together that are separated by a small gap), but it otherwise works very much the same and uses a small fraction of the RAM of FISHR2.

Table 1. Comparison of speeds in seconds across four IBD detection programs as a function of sample size in simulated chromosomes of length 16 cM and 1,185 SNPs.

Sample size	GERMLINE	FISHR2	HaploScore	rIBD
500	1.67	3.85	368.0	187.5
1k	6.13	14.5	1,468	1,406
2k	23.8	58.2	5,826	17,258
4k	92.3	225.6	23,134	39,724
8k	380	913	95,650	did not complete

8 Figures

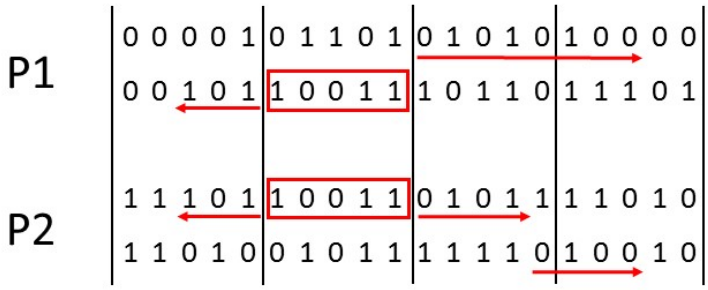

Figure 1: This figure displays the internal GERMLINE2 process of finding a potential segment and the extension process. The chromosomal segment is divided into smaller segments with a length defined by the `-bits` argument (in this case `-bits` 5) as denoted by the vertical lines. The perfect match in 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).

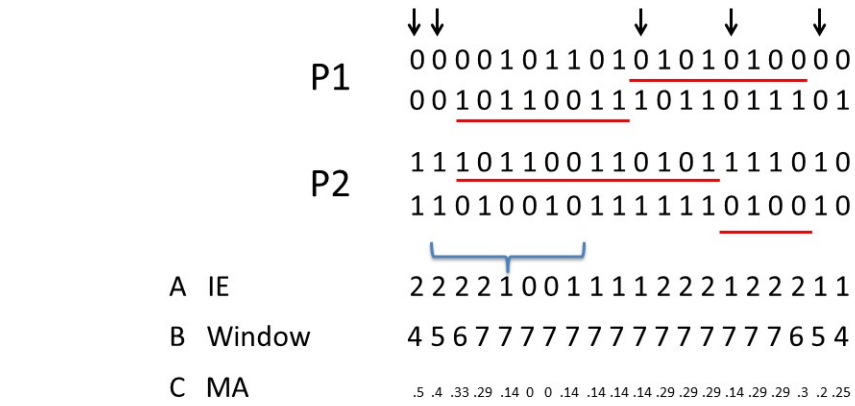

Figure 2: The moving average of IEs (MA) is calculated for each SNP within and around the called segment. In this example the `-window` value is 7, each SNP will use that SNP, as well as the three SNPs on either side of it, to calculate the MA value for that particular SNP.

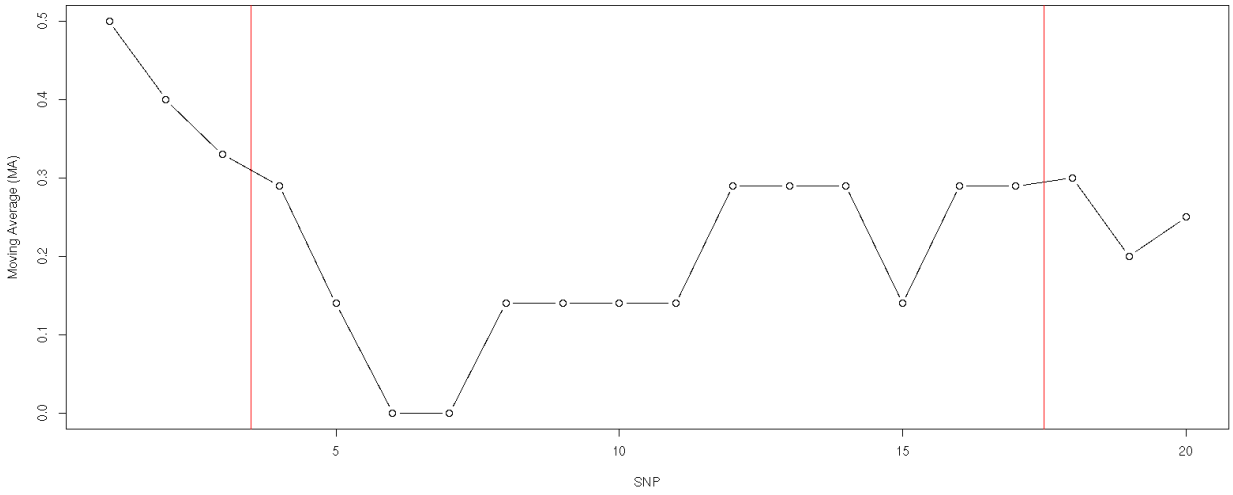

Figure 3: Once all the MA values are calculated, FISHR2 then starts in the center of the segment and move outward until a value is reached that is greater than the `-emp-ma-threshold` value. The point at which the MA for a SNP is above this value will determine the end points of the called segment. In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

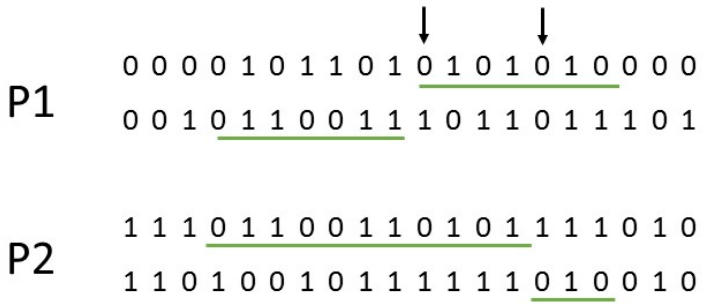

Figure 4: Once the endpoints are determined using the `-emp-ma-threshold`, the total proportion of SNPs that are IEs in the final segment is calculated. If this value is greater than that provided by the `-emp-pie-threshold`, the entire segment is dropped.

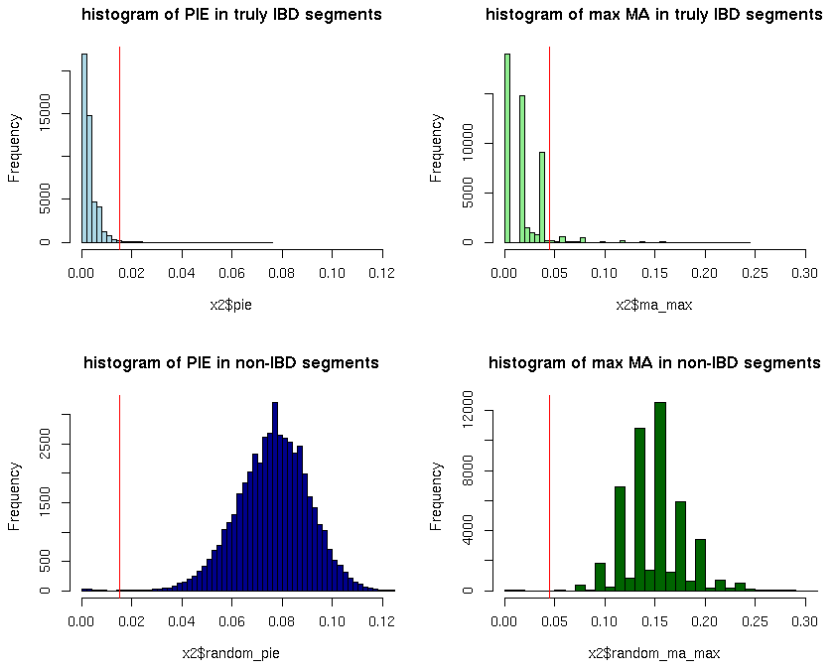

Figure 5: The top row shows the distributions of PIE and MA, respectively, in the middle 50% of very long segments of no opposite homozygotes, which are almost certainly IBD. Similarly, the bottom row shows the distributions of PIE and MA, respectively, in the middle 50% of segments at the same endpoints 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 (red vertical lines) 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 (red vertical lines) 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 segment endpoints, where IEs begin to accumulate, and therefore the estimates of false positive and false negative rates from these distributions is overly optimistic because it doesn't account for over- and under-extension of segments. 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.