iHDSel Manual v 0.5.4

Antonio Carvajal-Rodríguez
Facultad de Biología, Campus Lagoas-Marcosende
Departamento de Bioquímica Genética e Inmunología
Universidad de Vigo, Vigo 36310, Spain
Email: acraaj@uvigo.es
Web: http://acraaj.webs.uvigo.es

Table of Contents

Versions

Version iHDSel 0.5.4 (June 2026):

- All shared SNP positions are now considered candidate sites by default (-selectives inf).

- Added several LD-based block-construction modes (see the new section “LD-based block-construction modes”).

- Added support for monomorphic sites, which are ignored in LD calculations during block construction.

- In the command-line options, the -reference argument has been renamed to -focal-population, although -reference is still supported for backward compatibility.

- Improved the computational efficiency of the EOS outlier test by optimizing F_{ST} calculations.

- A minor bug affecting even values of -minwin has been fixed. Under certain circumstances, it could artifactually allow the acceptance of blocks exactly matching the specified even -minwin size.

Version iHDSel 0.5.3 (April 2026):

- Computational efficiency improvement: Identified and resolved a bottleneck in low-frequency SNP filtering across populations, substantially reducing runtime.

- The -selectives argument now supports inf, enabling computation of the J_{Hac} statistic across all shared positions.

- Fixed a bug in VCF reading that collapsed diploid genotypes into a single haplotype.

- Relaxed the requirement for VCF files to contain the same SNPs.

- Set doEOS to 0 (False) by default.

Version iHDSel 0.5 (May 2024):

Minor changes to improve speed.

Version iHDSel 0.4 (April 2024):

Calculation of blocks centered on outliers (-useblocks 0) has been slightly fixed to be done the same as -useblocks 1.

Version iHDSel 0.3 (April 2024):

Fixed a bug where haplotype blocks could be unusually long if there were some fixed SNPs in the reference population.

Version iHDSel 0.2 (April 2024):

Three changes from previous versions:

- The SNPs at the beginning and end of each block are now given in the output file.

- Under the -useblocks 0 option if mean F_{ST} is > 0.5 there is not outlier test (EOS) performed but the 90-percentile are used as centers of haplotpe blocks.

- New doEOS option (True by default) to allow the user to bypass the EOS test by setting -doEOS 0 in the command line arguments.

Version iHDSel 0.1 (November 2023):

This is a completely renewed version. The nvd method has been replaced by the J_{HAC} method which uses Jeffreys divergence (population stability index) to compare the distribution of haplotype classes between two populations. The minimum allele

frequency (MAF) is by default 0.01 and is computed for the metapopulation (both populations jointly) for the haplotype-based methods but for each population separately for the non-haplotype-based methods (see the corresponding section below).

Version HacDivSel 1.5 (June 2021):

- Corrects a minor bug that would provoke the program break if the major allele was a gap ('-') when reading the fastfasta format.

- Adds the possibility for the user of introducing specific positions as selective candidates via the argument -selectives (see the section of Candidates for selective loci).

- Minimum allele frequency (**MAF**): MAF changed to be a per population relative value (now by default 0.01) previously was an absolute metapopulation value (by default 4).

Introduction

iHDSel is a program for the detection of selection in pairs of populations that may be separated in space or time. It includes a new statistic, called J_{HAC}, which is based on the information theory interpretation of the Price equation (Frank 2012) and computes the Jeffreys divergence (population stability index) between the haplotype allelic class (Labuda *et al.* 2007, Hussin *et al.* 2010) distributions of both populations.

The J_{HAC} statistic have three main advantages for detecting divergent selection:

1.- Has a known distribution under the null hypothesis of no selection.

2.- It is fast because having a known distribution avoids the need for simulations or resampling to assign statistical significance.

3.- The statistic directly compares distributions, so each block is evaluated using a single statistical test. Therefore, no multiple-testing correction is required at the block level.

As with previous versions, the program also incorporates the extreme-outlier set (EOS) test (Carvajal-Rodríguez 2017).

The Program

Window size

No predefined window size

By default there is no predefined window size (-window 0) and the program has four different options to inspect candidates for selection and calculate the statistic over an haplotype window centered on each candidate.

1) Automatic overlapping blocks

The default option (-useblocks 1) instructs the program to construct LD-based haplotype blocks around candidate sites. By default, the minimum block size is 11 SNPs, although this value can be modified using the -minwin command-line argument; values of 5 or greater are allowed.

By default, all SNP positions shared between the two populations are considered candidate sites. A block is independently constructed around each candidate SNP, resulting in overlapping candidate-centered windows.

Under the default automatic block-construction mode, medium-g0 (-permissive-mode 3), block extension is based on pairwise LD between consecutive SNPs. Starting from a candidate SNP (c), the block is extended to the left and right while adjacent SNP pairs satisfy the required D' and r^2 criteria.

The default thresholds for this mode are:

$$D' \geq 0.4$$

and

$$r^2 \geq 0.01.$$

Thus, for a block containing (W) SNPs around candidate (c), the adjacent LD links evaluated during extension include:

$r^2_{(c-W/2,c-W/2+1)}, \ldots, r^2_{(c-1,c)}, r^2_{(c,c+1)}, r^2_{(c+1,c+2)}, \ldots r^2_{(c+W/2-1,\ c+W/2)}, \ldots$

Here, (r) is the correlation coefficient estimated from a sample of size (n), and D' is the normalized linkage disequilibrium coefficient (Lewontin 1964).

Starting with version 0.5.4, when the minimum block size is at least 11 SNPs, the r² condition is considered satisfied when either:

$r^2 \geq 0.01$ or $Pr(\chi_1^2 \geq nr^2) \leq \alpha$,

is significant at the selected significance level. The D' threshold must still be satisfied.

Under the default medium-g0 mode, no internal LD gaps are allowed. Alternative threshold levels, gap allowances, and average-based construction rules can be selected using -permissive-mode, as described in the section **LD-based block-construction modes**.

Block extension stops when the selected LD conditions are no longer satisfied. Consequently, a block meeting the minimum size cannot be constructed when there is insufficient detectable linkage disequilibrium around the candidate.

2) Automatic non overlapping blocks

If automatic block construction is desired but overlapping windows are not, the option `-selectives -1` can be used together with the default setting `-useblocks 1`.

In this mode, the program automatically constructs haplotype blocks as described above, but only one candidate site is assigned to each block. Consequently, the resulting windows are non-overlapping: each new block starts where the previous one ends, and the candidate site is placed at the center of the corresponding block.

3) Blocks around specific candidate sites

The user can propose candidate sites using the `-selectives` tag and the corresponding arguments in the command line to introduce the selective positions (see the section User-defined candidates for Selective Loci). The program uses the user-specified

positions as candidate sites to construct haplotype blocks, as described above. When the value is inf, all positions shared between the two populations are considered as candidates and we are in the same situation as in the Automatic overlapping blocks section.

4) Blocks around the outliers

The user can decide not to use the default option of automatic blocks simply by calling the program with the tag -useblocks 0. In this situation, the -doEOS option (disabled by default) must be activated (-doEOS 1). Two scenarios can happen here. First, if there are outliers the program will use them (extreme or not) and will construct the haplotype blocks around them as indicated above. Second, if there are no outliers, if the mean F_{ST} is greater than 0.5, it will use the 90th percentile values as candidates and again construct the haplotype blocks around those sites. If there are no outliers and the mean F_{ST} is not greater than 0.5, then the program will conclude that there are no candidates to build blocks and terminate.

Haplotype blocks constructed around outliers may be shorter than automatically constructed blocks. Therefore, the minimum permitted window or block size is lower in this case and can be set to as few as 3 SNPs using -minwin. The default block-construction mode is the same as for automatic blocks, namely medium-g0 (-permissive-mode 3), as described in the next section.

LD-based block-construction modes

Version 0.5.4 introduces several alternative rules for constructing LD-based blocks. The -permissive-mode argument accepts integer values from 0 to 17, defining block-construction modes that are broadly ordered from less to more permissive. Each mode corresponds to a particular combination of LD thresholds, allowed gaps, and construction rule.

The LD-based block-construction mode is described using the following naming convention:

<threshold level>-g<gap class>[-<non-default construction rule>]

Threshold level

The first component indicates the general threshold level (strict, medium, or permissive). The threshold levels used in the mode labels correspond to the following default values:

Threshold level	D' threshold	r^2 threshold
Strict	0.95	0.95
Medium	0.40	—
Light	0.20	0.01

There is no separate medium threshold for r^2. Modes using the medium D' threshold apply the light r^2 threshold under the standard link-wise rule, whereas the r^2 criterion is inactive under the average-based rule.

For windows shorter than 11 SNPs, the r^2 condition is evaluated by direct comparison with the selected threshold. For windows containing at least 11 SNPs, the r^2 condition is considered satisfied when either r^2 reaches the selected threshold or the χ^2 test based on (nr^2), with one degree of freedom, is significant at the selected significance level. In all cases, the D' condition must also be satisfied.

Gap class

The gap component indicates the number of internal LD gaps allowed during block construction:

- g0: no internal LD gaps are allowed on either side of the candidate.
- g1: up to one internal LD gap is allowed per side.
- g2plus: the per-side allowance starts at two internal LD gaps and may increase up to a maximum of 10, depending on the minimum window size.

For candidate-centered blocks, gap limits are applied independently to the left and right sides of the candidate. Therefore, a block may contain up to twice the per-side limit.

For g2plus modes, the per-side gap allowance is calculated as:

```
min(maxgaps, max(mingaps, minblock / step))
```

using integer division, with mingaps = 2 and maxgaps = 10. The value of step is 30 under the standard link-wise rule and 10 under the avgDprime rule. In both cases, the maximum number of consecutive LD gaps is fixed at one.

Table 1. LD-gap tolerance under the g2plus mode as a function of the minimum window size (minwin).

Allowed gaps per side	minwin, link-wise rule (step = 30)	minwin, avgDprime rule (step = 10)	Maximum total gaps
2	3–89	3–29	4
3	90–119	30–39	6
4	120–149	40–49	8
5	150–179	50–59	10
6	180–209	60–69	12
7	210–239	70–79	14
8	240–269	80–89	16
9	270–299	90–99	18
10	≥300	≥100	20

Construction rule

When no construction-rule suffix is shown, blocks are constructed using the standard link-wise rule. Under this rule, each LD link must satisfy both the D' and r^2 criteria, except for the permitted number of gaps.

For windows shorter than 11 SNPs, the r^2 condition is evaluated by direct comparison with the selected threshold. For windows containing at least 11 SNPs, the r^2 condition is satisfied when

either r^2 reaches the selected threshold or the χ^2 test based on (nr^2), with one degree of freedom, is significant at the selected significance level. In all cases, the *D*' condition must also be satisfied.

The avgDprime suffix identifies the alternative average-based construction rule. In this mode, the r^2 criterion is inactive, and block acceptance depends only on whether mean *D*' reaches the threshold associated with the selected threshold level.

For automatic non-overlapping blocks (-useblocks 1 -selectives -1), modes 9–17 are interpreted using the corresponding standard link-wise rule rather than the average-based rule. In this context, the r^2 criterion is reactivated: the strict r^2 threshold is used for strict modes, whereas the light r² threshold is used for medium and permissive modes. In addition, LD gaps are currently disabled. Therefore, the effective gap class is always g0, regardless of the gap class associated with the selected -permissive-mode value. Modes within each group of three consequently produce the same effective configuration in this context. Thus, modes 9-11 are reported as strict-g0, 12-14 as medium-g0, and 15-17 as permissive-g0.

The default block-construction mode is medium-g0, equivalent to -permissive-mode 3, for both automatic block construction and outlier-centered LD-based blocks.

Modes	Candidate-centered mode label	D' threshold	r² threshold	Candidate-centered construction rule	Effective rule for automatic non-overlapping blocks
0–2	strict-g*	Strict	Strict	Link-wise AND	strict-g0
3–5	medium-g*	Medium	Light	Link-wise AND	medium-g0
6–8	permissive-g*	Light	Light	Link-wise AND	permissive-g0
9–11	strict-g*-avgDprime	Strict	Inactive	Mean D'	strict-g0
12–14	medium-g*-avgDprime	Medium	Inactive	Mean D'	medium-g0

Modes	Candidate-centered mode label	D' threshold	r^2 threshold	Candidate-centered construction rule	Effective rule for automatic non-overlapping blocks
15–17	permissive-g*-avgDprime	Light	Inactive	Mean D'	permissive-g0

Within each group of three modes, g* corresponds successively to g0, g1, and g2plus.

Predefined window size

The window size can be defined by the user by the command line tag -window. The size of the window must be at least the minwin value (default 11). There are two options now:

1) There are not selective sites defined by the user. Then, the program applies the -usebloks 0 option as was explained in the previous section (option 4, Blocks around the outliers).The blocks are discarded only if they do not meet the minimum condition of having an average *D*'>0.

2) There are user-defined selective sites. The program centers each candidate site in a window with the predefined size.

If the window size is predefined using -window, the -useblocks 1 option is not allowed, because there is no need to construct LD-based blocks when the window size is already specified.

Download

The program can be downloaded from

https://acraaj.webs.uvigo.es/iHDSel.html

Ready-to-use executables

Windows 64 bits: iHDSel0.5.4.exe

Ubuntu: iHDSel0.5.4_U

macOS: iHDSel0.5.4_M3Pro

Compiling the program manually (linux/macOS)

In a terminal window (console) go to the source folder that should include a file called Makefile, and just type make.

Input

The acceptable formats for the input file are VCF (see the VCF subsection below), MS, Fasta, HapMap phased, FastPhase, Genepop, BayeScan (codominant markers) and PLINK. The three last formats allow only for the execution of the outlier-based method (EOS).

The VCF format must begin with ##fileformat tag. For the remaining formats iHDSel accepts as commentaries those lines in the very initial part of the file beginning with #.

Additionally, under the formats Genepop, HapMap and PLINK, any line in any part of the file can be a commentary if it begins with #. Therefore, for these files we can comment some lines in order to eliminate specific individuals (Genepop and PLINK-ped) or SNPs (HapMap) from the sample.

VCF format

The new version iHDSel accepts a restricted subset of the VCF format. To use this format:

```
–format vcf
```

An example of a program calling

```
./iHDSel -input1 snps_rads.chr5.phased_A.vcf -input2
snps_rads.chr5.phased_B.vcf -format vcf -output iHDSv0_Chr5 &
```

or the equivalent alternative with explicit default values

```
./iHDSel -path ./ -input1 snps_rads.chr5.phased_A.vcf -input2
snps_rads.chr5.phased_B.vcf -format vcf -output iHDSv0_Chr5 -verbose 0 -minwin
11 -maf 0.01 -useblocks 1 &
```

VCF header

The header must begin with the ##fileformat tag and the last header line must include the FORMAT tag before the sample names, for example if there are two samples the last header line could be:

```
#CHROM POS ID REF ALT QUAL FILTER INFO FORMAT sample1 sample2
```

The columns of a VCF

The chromosome must be the same within and between the files of the two populations. Likewise, the positions must coincide in both populations to be analyzed.

The format type must be GT (genotypes) and phased e.g. 0|0 or 0|1 or 1|1.

For example a valid file for fhe first population with 4 SNPs and 3 samples could be:

```
##fileformat=VCFv4.2
##FILTER=<ID=PASS,Description="All filters passed">
##source=shapeit4.1.3
##contig=<ID=5>
##INFO=<ID=AF,Number=A,Type=Float,Description="Allele Frequency">
##INFO=<ID=AC,Number=1,Type=Integer,Description="Allele count">
##FORMAT=<ID=GT,Number=1,Type=String,Description="Phased genotypes">
#CHROM POS    ID     REF    ALT    QUAL   FILTER INFO   FORMAT S1     S2     S3
5      243588 .      C      A      .      .      AC=3;AF=0.0194805;CM=0     GT
       0|0    0|0    0|0
5      254234 .      T      G      .      .      AC=1;AF=0.06935;CM=0.06     GT
       0|0    0|0    0|0
5      261675 .      T      G      .      .      AC=3;AF=0.01805;CM=0.01     GT
       0|0    0|0    0|0
5      420619 .      G      T      .      .      AC=13;AF=0.0846;CM=0.177 GT
       0|0    0|0    0|0
```

and for the second populations with the same 4 SNPs and 2 samples:

```
##fileformat=VCFv4.3
#CHROM POS    ID     REF    ALT    QUAL   FILTER INFO   FORMAT S7     S15
5      243588 .      C      A      .      .      AC=3;AF=0.004805;CM=0      GT
       0|1    0|0
5      254234 .      T      G      .      .      AC=1;AF=0.04351;CM=0.010646
       GT     0|0    1|0
```

```
5       261675 .       T       G       .       .       AC=3;AF=0.00805;CM=0.018087
        GT     0|0     0|0
5       420619 .       G       T       .       .       AC=13;AF=0.0156;CM=0.177031
        GT     1|0     0|0
```

MS format

The MS format is the standard Hudson's ms output format. For example:

```
ms 2 2 -t 60

104642148


//

segsites: 3

positions: 0.002771     0.004018    0.014038

000

111


//

segsites: 3

positions: 0.1    0.4   0.8

010

101
```

To use this format (is the default so no argument is needed at all):

```
–format ms
```

We can read two runs from the same file as if they were different populations (the example above) as produced in the output of the old GenomePop software for samples of two subpopulations. In this case the sample size is assumed to be the same for both populations. For example, if we have generated

the file called mu_r_60Nm_10.txt that has 50 haplotypes sampled from each population; to analyze it under iHDSel we type:

```
./iHDSel -path ./ -input mu_r_60Nm_10.txt -output test.out  -sample 50 -format ms -maf 0.01
```

Alternatively, we could has just one run in the file that is a mixture of two populations as produced by using the corresponding arguments when executing the MS program

```
./ms 100 1 -t 60 -r 60 1000000 -I 2 55 45 40 >> mu_r_60Nm_10.txt
```

that produces one run with a sample of 100 sequences: 55 coming from one population and 45 from another. Then we get, for example:

```
ms 100 1 -t 60 -r 60 1000000 -I 2 55 45 40

104642148


//

segsites: 300

positions: 0.002771     0.004018    0.014038 ...

000 ...

111 ...

110 ...

010 ...

.

.

000 ...
```

In this case if we want that iHDSel reads the file we should indicate both sample sizes

```
./iHDSel -path ./ -input mu_r_60Nm_10.txt -output test2.out  -sample 55 -sample2 45 -format ms -maf 0.01
```

FASTA format

The program reads the FASTA sequential format. It allows symbols A, C, G and T. Entries with any other symbol represent unresolved nucleotides and are treated as null alleles. When

preparing the data for the computation, the following parsing is performed. In the reference population (the first, by default under equal sample sizes or the one with the largest sample size), the most frequent allele at each position will be noted as '0' and any alternate as '1'. If the most frequent symbol is N that site is ignored. Thus, if the most frequent allele in the reference population is, say, T then this base will be noted as '0' and any other as '1' for this site in the whole data set (all populations).

To use this format:

```
–format fasta
```

The two populations can be read from the same file or from separate files.

1) From separate files

```
./iHDSel -path ./TESTS/ -input1 filt_EN3.fas -input2 filt_SA.fas -format fasta
-output filt_EN3_SA -maf 0.01 &
```

2) From one file with two samples with equal sample size

```
./iHDSel -path ./TESTS/ -input EN3_SA.fasta -format fasta -output EN3_SA_mix -
maf 0.01 &
```

In this case the program assumes the file EN3_SA.fasta have half of the sequences from and sample and half from the other. If that is not the case the following option should be used.

3) From one file with two unequal samples

```
./iHDSel -path ./TESTS/ -input EN3_SA.fasta -format fasta -output EN3_SA -maf
0.01 -tag ENGLAND &
```

Note the use of `-tag ENGLAND` to separate sequences that include that word in the name from other sequences that do not. The sample carrying the tag in the sequences name will be considered the first sample.

There is also a tag -save to store the obtained samples in the corresponding files.

```
./iHDSel -path ./TESTS/ -input1 filt_EN3.fasta -input2 filt_SA.fasta -format
fasta -output filt_EN3_SA -maf 0.01 -save 1 &
```

FastPHASE 1.4 format

To use this format:

```
–format fp
```

The program expect two files, one for each population (can have different sample sizes). The SNPs in both files must be the same in the same order. That is, if the i-th SNP in population-1 is rs11901199 then rs11901199 is also expected in the i-th position in the population-2.

To analyze the files F1_hapguess_switch and F2_hapguess_switch that are in a folder called FastPhase, we use the following arguments:

```
./iHDSel -path ./FastPhase/ -input F1_hapguess_switch -input2
F2_hapguess_switch -output hapguess_ -format fp
```

HapMap phased format

Files are organized in a SNPs x haplotypes format, so each row represents a SNP and each column a haplotype. The first row (header row) contains the identifiers of the individuals that have been phased. The first column (header column) contains the rsID of the SNPs. The second column contains the physical position of these SNPs in the particular chromosome. Entries with anything other than A, C, G, or T represent unresolved nucleotides and are treated as null alleles. To use this format:

```
–format hm
```

The program expect two files, one for each population (can have different sample sizes). To make the program read one population from each file we need to add the argument –singlepopfile 1. For example, to analyze the files hm3_chr2_ceu.txt and hm3_chr2_yri.txt that are in a folder called HapMap, we use the following arguments:

```
./iHDSel -path ./HapMap/ -input hm3_chr2_ceu.txt -input2 hm3_chr2_yri.txt -
output hmCEUvsYRI -format hm -singlepopfile 1
```

Genepop format

This format has no haplotype information thus, only the EOS test is performed using the GST estimator (Nei 1973). It allows variable sample size as inferred from the data.

To use this format:

```
–format gp
```

```
./iHDSel -path ./ -input mu_r_60Nm_10.genepop -output test.out   -format gp -maf 0.01
```

BayeScan format

This format has no haplotype information thus, only the EOS test is performed using the GST estimator. The version of the data format is for codominant markers. It assumes equal sample size in both populations. For example if the number of copies is 2N = 100 we set -sample 50. To use this format:

```
–format bs
```

```
./iHDSel -path ./ -input mu_r_60Nm_10.bs -output test.out   -format bs -sample 50  -maf 0.01
```

PLINK flat files (MAP/PED) format

This format has no haplotype information thus, only the EOS test is performed using the GST estimator. It allows variable sample size which is inferred from the data.

MAP files

The fields (separated by space or tab) in a MAP file are:

- Chromosome
- Marker ID
- Genetic distance
- Physical position

For example:

```
2     rs11901199  0     8856
2     rs7594567   0     11833
```

The column with the genetic distances is optional. The following example is also valid:

```
2     rs11901199  8856
2     rs7594567   11833
```

Thus, the number of columns must be 4 or 3, a mixture of both is not valid.

PED files

The fields (separated by space or tab) in a PED file are

- Population ID
- Individual ID
- Paternal ID
- Maternal ID
- Sex (1=male; 2=female; other=unknown)
- Phenotype (0=unknown; 1=unaffected; 2=affected)
- Genotypes (space or tab separated, 2 for each marker)

For example, corresponding to the map file above, the first line could be:

```
Pop1  NA19176     0 0 1 1      G G C C
```

…

Note that the length of the genotype must be 2x the number of data rows in the MAP file.

The genotypes can be nucleotides or numbers. In the case of nucleotides there are only 4 alleles (A, C, G or T) and any other symbol will be considered as missing information (N).

To use this format with nucleotides:

```
–format pl
```

In the case that genotypes are numbers, the maximum number of alleles allowed in the current version is 255 and 0 is the symbol for a missing allele.

To use this format with numbers:

```
–format pl2
```

In any case, if the name of map and ped files is the same e.g. mu_r_60Nm_10.map and mu_r_60Nm_10.ped then the input file for the program must be

```
-input mu_r_60Nm_10
```

without the extension.

For example, for nucleotides:

```
./iHDSel -path ./ -input mu_r_60Nm_10 -output test.out   -format pl
```

Or for numbers:

```
./iHDSel -path ./ -input mu_r_60Nm_10 -output test.out   -format pl2 -maf 0.01
```

That read both the map and ped files. However, if the names of map and ped files are different e.g. mu_r_60Nm_10.map and mu_r_60Nm_10_nuc.ped then the input for the program must be

```
-input mu_r_60Nm_10 -input2 mu_r_60Nm_10_nuc -singlepopfile 1
```

again without the extension and with the flag singlepopfile to indicate that two different file names are being considered. Note that in this latter case the -input2 name must be always the ped file name. For example, with nucleotides:

```
./iHDSel -path ./ -input mu_r_60Nm_10 -input2 mu_r_60Nm_10_nuc -singlepopfile
1   -format pl
```

User-defined candidates for selective loci

For any haplotype-based input format, one or more SNPs can be specified as candidate sites for selection. Since not all supported formats include SNP identifiers, candidate sites must generally be specified by their position within the haplotype. The only exception is the special value inf (see below), which instructs the program to consider all shared SNP positions as candidate sites.

For FASTA and fastPHASE formats, SNP positions are 0-indexed (i.e. the first position is 0, not 1). Thus, in a haplotype containing 1,000 SNPs, positions range from 0 to 999. For example, a SNP located at position 700 in 1-based indexing corresponds to position 699 in 0-based indexing, and should be specified as:

```
-selectives 699
```

Multiple candidate SNPs must be provided in increasing order, for example:

```
-selectives 1 699 700
```

As an illustration, consider haplotypes from different populations of Atlantic salmon obtained using fastPHASE. Suppose we want to test a SNP located in the NADP-dependent malic enzyme-2 locus (mMEP-2*), for which independent evidence suggests local adaptation. This SNP (ctg7180001794010_7928_SCT) is located at 39,897,822 bp on chromosome Ssa25 and is flanked by two SNPs at −11 Kb and +15.6 Kb.

After filtering the microarray data, the SNP of interest is located at position 3579 (0-based indexing), with the flanking SNPs at positions 3578 and 3580. Additionally, another SNP at 19,827,046 bp (position 2118) is included as a control, as previous analyses indicate a J_{HAC} value close to 0.

The corresponding command would be:

```
-path /home/workdir/ -input1 atl_25_hapguess_switch.out -input2
can_25_hapguess_switch.out -format fp -output nvdfst_Atl_Can_25_MEPs2_2118 -
selectives 2118 3578 3579 3580
```

For the MS format, SNP positions are relative and range from 0 to 1. For VCF and HapMap formats, SNP positions correspond to the physical coordinates given in the second column of the input files.

Special value: inf (all shared SNPs)

When the value inf is provided to -selectives, all SNP positions shared between the two populations are automatically considered as candidate sites. This allows computing the J_{Hac} statistic

genome-wide over the full set of shared positions, without manually specifying individual SNPs. This is the default behavior of the program when no candidate sites are explicitly specified.

Reference haplotype in spatial and temporal comparisons: the focal population

The J_{HAC} method is based on calculating the distribution of HAC values in each sample. The HAC value is calculated as the Hamming distance of each haplotype with respect to a reference configuration (Carvajal-Rodríguez 2017, 2024). This reference configuration is calculated for one of the samples which we refer to as the focal population.

The focal population is the population used to define the HAC reference haplotype and to construct LD-based blocks. When a target population for the sweep is known, it is natural to use that population. Otherwise, the largest sample may provide a more stable estimate of the reference haplotype.

By default, the focal population is initially chosen as the largest sample or, if both samples have the same size, the first sample. However, when the focal population is left undefined, the program may also use the other sample as the focal population if no suitable blocks can be found in the initially selected sample.

Alternatively, the focal population can be specified using the -focal-population option with a value of 1 or 2, indicating the first or second sample, respectively.

Therefore, for samples T1 and T2, where T2 corresponds to the more recent time point (e.g. tagged as 2021-03), the command

```
 -path /home/workdir/ -input1 T1.fas -input2 T2.fas -format fasta -output T1vsT2
```

will use:

i) The first sample (T1) as the focal population if both samples have the same size, or

ii) The largest sample if the sample sizes differ.

In addition, for both cases (i) and (ii), if no valid block can be constructed around a candidate site in the initially selected focal population, the program will also evaluate that site using the other sample as the focal population.

Thus, under automatic focal-population selection, the program may evaluate candidate sites in both samples: first in the automatically selected focal population and, when no valid block can be constructed, in the alternative sample.In the output file name, this behavior is indicated by the text AutoFocalPopX, where X identifies the automatically selected focal population.

In contrast, the command

```
-path /home/workdir/ -input1 T1.fas -input2 T2.fas -format fasta -output T1vsT2ref1 -focal-population 1
```

uses only T1.fas as the focal population for block construction. In the output file name, this is indicated by the text FocalPop1.

Similarly,

```
-path /home/workdir/ -input1 T1.fas -input2 T2.fas -format fasta -output T1vsT2ref2 -focal-population 2
```

uses only T2.fas as the focal population for block construction. In the output file name, this is indicated by the text FocalPop2.

In summary, under the automatic block construction option (-useblocks 1), the program uses the specified focal population or, if none is specified, starts with the largest sample. If no valid block can be constructed around a candidate site in that sample, the program attempts block construction using the other sample as the focal population.

Run the program

Command line options

The following arguments can be passed to the program:

-cutoff<DOUBLE> Define the value numdesvs in the computation of the cutoff for the EOS test (cutoff = fsttot+numdesvs*maxdev). Default is 1.25.

-doEOS <INTEGER> From version 0.5.3, this option defaults to 0. When set to 0, the EOS test is not performed. If the input format does not involve haplotypes and/or the useblocks option (see below) is set to 0, doEOS must be set to 1.

-focal-population <INTEGER> The population in which the selective sweep is assumed to occur and which is used as the reference population for computing haplotype allelic classes (HACs). Valid values are 1 or 2, corresponding to the first and second input samples, respectively. By default, no focal population is specified. In this case, the program initially uses the largest sample as the focal population (or the first sample if both samples have the same size) and may switch to the other sample when block construction is not possible in the initially selected focal population. Previous versions used the -reference tag for the same purpose; this tag remains available for backward compatibility.

-help|-h Displays this help message.

-input <STRING> Specifies the name of the input file.

-input2 <STRING> Specifies the name of the second input file. It is mandatory only when the format is vcf or the argument -singlepopfile is set to 1.

-format <STRING> Specifies the sequence data format (BS, Fasta, VCF, GP, HM, PL or, by default, MS).

-maf <DOUBLE> Defines the minimum allele frequency in each population (default is 0.01).

-maxwin <INTEGER> Defines the maximum window size (default is 1000) under the variable window size setting. If set to 0 then it defines all available SNPs as the maximum window size.

-minwin <INTEGER> Defines the minimum block size (default is 11 SNPs).

-nogaps <INTEGER> By default is 0. Only available under the Fasta format. When set to 1 it skips columns carrying gaps.

-onlyEOS <INTEGER> Defines whether only the EOS test is performed. Default is 0 (J_{HAC} and EOS performed). It is automatically set to 1 when the format is BS, GP or PL.

-onlyrsidInfo <INTEGER> By default is 0. When set to 1 the program generates an output file with the names, positions and frequencies of the set of SNPs.

-output <STRING> Specifies the header for the name of the output file.

-path <STRING> Specifies the path to the input file.

-permissive-mode <INTEGER> Selects the LD-based block-construction mode (integer from 0 to 17), which determines the LD thresholds, gap allowance, and construction rule. The default is mode 3 (medium-g0). See “LD-based block-construction modes” for details.

-reference <INTEGER> Legacy synonym of -focal-population. This option is retained for backward compatibility and behaves identically to -focal-population.

-sample <INTEGER> Defines the number of alignments in each population sample (50 by default). It is required under the MS and BayeScan formats. It is not required Under the HM, VCF, Fasta and FastPhase formats because the two files can have different sample sizes that will be recognized automatically. If under the MS format we want to read two populations with different sample sizes we need to add a second argument –sample2 (see below). Under the GenePop format the sample size is allowed to vary between populations. Under the BayeScan format the size is implicitly allowed to vary (total number of genes / ploidy) but the value of sample will be used to compute the minimum allele frequency.

-sample2 <INTEGER> Defines the number of alignments in a second population sample under the MS format. By default has value 0, in this case the program reads two populations (marked with //) from one MS file and expects that each population has the same sample size. If sample2 has a positive value then the program expects a second input file name (-input2 …) and would read each population in MS format from a different file: -path ./msfiles/ -input msfile1 -input2 msfile2 -output testmsf -sample 40 -sample2 55 -format ms.

-save <INTEGER> Default: 0. When using FASTA format and both populations are stored in the same input file and identified by different tags, setting -save 1 causes the separated samples to be written to independent output files.

-selectives <DOUBLE> User defined candidate sites to look for selection (if there are more than one value they must be separated by

spaces).When the value is inf, all positions shared between the two populations are considered as candidates.

-singlepopfile <INTEGER> This is only necessary under the HM or PLINK formats. By default is 0 so it is assumed that there are two populations (with equal sample size) at each HM input file. When set to 1 this means that two different filenames (-input and -input2, one for each population) should be provided for the program to work. In the case of PLINK this is used for reading map and ped files with different names (see PLINK format section). In the case of the VCF or fastphase formats it is not necessary since the format already require two files (one per population).

-SL <DOUBLE> Defines the significance level (α=0.05 by default).

-tag <STRING> When the format is Fasta and we read one file with different sample sizes, it permits to look for the given string in the sequence names to separate the data in two samples.

-useblocks <INTEGER> By default is 1. It computes the haplotype blocks with the selective candidates in the middle as indicated in the corresponding section. When set to 0, the program uses the SNPs as the center of the blocks that are computed in a similar way.

-verbose<INTEGER> Provides additional information during the computation and a log file at the end.

-window <INTEGER> Defines the fixed window size. If 0 the window size will be variable (this is the option by default).

Default Values

Calling the program without arguments, i.e.

```
./iHDS0.5.4
```

It is equivalent to calling

```
./iHDS0.5.4  -path ./ -format ms -input iHDSF -input2 iHDSF -window 0 -sample 50 -sample2 0  -maf 0.01 -cutoff 1.25 -output  iHDS_out -SL 0.05 -minwin 11 -maxwin 1000 -window 0 -onlyEOS 0 -doEOS 0  -singlepopfile 0 -onlyrsidInfo  0 -useblocks 1 -selectives inf -save 0 -nogaps 0 -verbose 0
```

Minimum allele frequency

The J_{HAC} method only considers SNPs that have at least a frequency of MAF (by default 0.01) when considering jointly both populations.

However, for the non haplotype formats (BS, GP or PL) it is required that the frequency of the allele to be at least MAF at each population separately.

Run in MacOsX or Linux

If the downloaded executable has name iHDSel (can vary for different versions), from the console or terminal just type ./iHDSel jointly with the desired arguments.

Run in Windows

Double click just for running the program under default options. Alternatively, you can go to the command prompt (cmd.exe) and type

iHDSel

You can also access the Run command by pressing the Windows logo key  +r

then drag and drop the .exe file from your folder and add the desired arguments, e.g. if the iHDSel.exe is in the folder work then after drag and drop you will have

```
C:\work\iHDSel.exe
```

now add the desired arguments and then hit the Intro key. For example:

```
C:\work\iHDSel.exe -input neutral.txt -output neutralres
```

Usage examples

The following examples assume that the terminal/command prompt is located in the directory containing the iHDS0.5.4 executable, and that the input data are stored in the data folder specified in the -path argument.

To analyze files in VCF format using blocks and considering all sites as candidates (i.e. -selectives inf), without including the EOS test, the command line could be:

Windows

```
iHDSv0.5.4.exe -path C:\Users\PC\data\ ^
-input1 data_pop1_chr2.vcf ^
-input2  data_pop2_chr2.vcf ^
-output pop1vspop2_chr2 ^
-format vcf  -verbose 0 -minwin 11 -maf 0.05 &
```

Linux/MacOS

```
./iHDS0.5.4 -path /home/data/ \
-input1 data_pop1_chr2.vcf \
-input2  data_pop2_chr2.vcf \
-output pop1vspop2_chr2 \
-format vcf  -verbose 0 -minwin 11 -maf 0.05 &
```

Output

The program generates three files. For example, if the ouput tag in the argument list was followed by X (`-output X`) then the program generates one file called X_MAF_0.01_Blocks_FSTs.txt with the result of the EOS analysis and other two called X_MAF_0.01_Blocks.tsv and X_MAF_0.01_Blocks_SIG.tsv with the result of the J_{HAC} analysis.

J_{HAC} result

The .tsv files would contain the following columns:

File: The name(s) of the analyzed file(s).

#SNPs: The number of SNPs analyzed.

Candidate: The candidate number if more than one were considered.

SNP: Position (0-indexed when not VCF or hapmap) of the candidate SNP.

Wide: The window size in number of SNPs.

Ini: The initial SNP of the haplotype block.

End: The terminal SNP of the haplotype block.

NCLASSES: The number of classes in which the data were binned to calculate the J_{HAC} statistic.

[1]**MeanFST:** The average value of F_{ST} in the analyzed data.

[1]**FST:** The F_{ST} value at the candidate site.

[1]**FSTIndex:** For a given candidate i the F_{ST}_i minus the average F_{ST} value.

[1]**fstPval:** The p-value after performing the Lewontin and Krakauer (LK) test.

n1: Sample size of the first population.

n2: Sample size of the second population.

JHac: Value of the J_{HAC} statistic.

Pval: The p-value for the J_{HAC} statistic value.

Sig: Indicates the significance. ns: non-significant. *: significant (Pval<SL).

[1] Available only when `-doEOS 1` **is enabled.**

EOS result

The .txt file would contain the list of F_{ST} outliers. First indicates the non-EOS outliers and after them the EOS outliers.

For each type of outliers the first line indicates the file names, the number of SNPs and the number of the given outliers. After that, a table follows with the following columns:

ID: The name of the SNP or locus or the position if a name is not available.

Pos: The position (0-indexed) of the outlier.

FST: The F_{ST} or G_{ST} value.

pval: The p-value after the LK test.

References

Carvajal-Rodríguez A. HacDivSel: Two new methods (haplotype-based and outlier-based) for the detection of divergent selection in pairs of populations. *PLoS One* 2017;**12**(4):e0175944. 28423003. https://doi.org/10.1371/journal.pone.0175944.

Carvajal-Rodríguez A. iHDSel software: The price equation and the population stability index to detect genomic patterns compatible with selective sweeps. An example with SARS-CoV-2. *Biology Methods and Protocols* 2024;**9**(1):bpae089. https://doi.org/10.1093/biomethods/bpae089.

Frank SA. Natural selection. V. How to read the fundamental equations of evolutionary change in terms of information theory. *J Evol Biol* 2012;**25**(12):2377–96. 23163325.

Hussin J, Nadeau P, Lefebvre JF *et al.* Haplotype allelic classes for detecting ongoing positive selection. *BMC Bioinformatics* 2010;**11**(1):65. doi:10.1186/1471-2105-11-65.
Labuda D, Labbé C, Langlois S *et al.* Patterns of variation in DNA segments upstream of transcription start sites. *Hum Mutat* 2007;**28**(5):441–50. https://doi.org/10.1002/humu.20463.
Lewontin RC. The Interaction Of Selection And Linkage. Ii. Optimum Models. *Genetics* 1964;**50**:757–82. 14221879.
Nei M. Analysis of gene diversity in subdivided populations. *Proceedings of the National Academy of Sciences* 1973;**70**(12):3321–3.