`selscan` v 1.1.0 User Manual

Zachary A Szpiech

May 11, 2015

Contents

1 Introduction

Extended Haplotype Homozygosity (EHH) (Sabeti *et al.*, 2002), Integrated Haplotype Score (iHS) (Voight *et al.*, 2006), Cross-population Extended Haplotype Homozygosity (XPEHH) (Sabeti *et al.*, 2007), and nSL (Ferrer-Admetlla *et al.*, 2014) are statistics designed to use phased genotypes to identify putative regions of recent or ongoing positive selection in genomes. They are all based on the model of a hard selective sweep, where a *de novo* adaptive mutation arises on a haplotype that quickly sweeps toward fixation, reducing diversity around the locus. If selection is strong enough, this occurs faster than recombination or mutation can act to break up the haplotype, and thus a signal of high haplotype homozygosity can be observed extending from an adaptive locus. These statistics, nSL in particular, retain some power to detect soft sweeps as well.

As genetics data sets grow larger both in number of individuals and number of loci, there is a need for a fast and efficient publicly available implementation of these statistics. Below we introduce these statistics and provide concise definitions for their calculations.

When using `selscan` please cite Szpiech and Hernandez (2014) as well as the appropriate paper that introduced the statistic: EHH (Sabeti *et al.*, 2002), iHS (Voight *et al.*, 2006), XP-EHH (Sabeti *et al.*, 2007), nSL (Ferrer-Admetlla *et al.*, 2014).

This introduction has been adapted from Szpiech and Hernandez (2014).

2 Obtaining `selscan`

`selscan` pre-built binaries and source code are available at `https://github.com/szpiech/selscan`. Binaries have been compiled on OSX 10.8.5, Ubuntu 12.04 (LTS), and Windows 7, but they should function across most versions of these operating systems. To compile from source, change directories to the `src/` directory and type `make`. Some minor modification (commenting and uncommenting certain lines) to the Makefile may be necessary depending on your target OS. `selscan` depends on the POSIX standard threading library (pthreads) and the zlib library (`http://www.zlib.net/`). A win32 implementation of pthreads is available at `https://www.sourceware.org/pthreads-win32/`, and a win32 implementation of zlib is available at `http://gnuwin32.sourceforge.net/packages/zlib.htm`. The windows version of `selscan` was built using a MinGW environment (`http://www.mingw.org/`), although it should only be necessary to set this environment up if you wish to compile from source on Windows.

The companion program `norm` depends on GNU GSL (`http://www.gnu.org/software/gsl/gsl.html`), and precompiled static libraries for each OS are included in the source code.

3 Basic Usage

To calculate EHH:

```
selscan --ehh <locusID> --hap <haplotypes> --map <mapfile> --out <outfile>
```

To calculate iHS:

```
selscan --ihs --hap <haplotypes> --map <mapfile> --out <outfile>
```

To calculate XP-EHH:

```
selscan --xpehh --hap <pop1 haplotypes> --ref <pop2 haplotypes> --map <mapfile> --out
<outfile>
```

To calculate nSL:

```
selscan --nsl --hap <haplotypes> --map <mapfile> --out <outfile>
```

4 Statistics implemented

Here we describe the various statistics implemented in `selscan`. Sections 4.1, 4.2, and 4.3 are reproduced with minor modifications from Szpiech and Hernandez (2014).

4.1 Extended Haplotype Homozygosity (EHH)

Extended Haplotype Homozygosity (EHH) was introduced by Sabeti *et al.* (2002). In a sample of n chromosomes, let $\mathcal{C}$ denote the set of all possible distinct haplotypes at a locus of interest (named x_0), and let $\mathcal{C}(x_i)$ denote the set of all possible distinct haplotypes extending from the locus x_0 to the i^{th} marker either upstream or downstream from x_0. For example, if the locus of interest x_0 is a biallelic SNP where 0 represents the ancestral allele and 1 represents the derived allele, then $\mathcal{C} := \{0, 1\}$. If x_1 is an immediately adjacent marker, then the set of all possible haplotypes is $\mathcal{C}(x_1) := \{11, 10, 00, 01\}$.

EHH of the entire sample, extending from the locus x_0 out to marker x_i, is calculated as

$$EHH(x_i) = \sum_{h \in \mathcal{C}(x_i)} \frac{\binom{n_h}{2}}{\binom{n}{2}}, \tag{1}$$

where n_h is the number of observed haplotypes of type $h \in \mathcal{C}(x_i)$.

In some cases, we may want to calculate the haplotype homozygosity of a sub-sample of chromosomes all carrying a 'core' haplotype at locus x_0. Let $\mathcal{H}_c(x_i)$ be a partition of $\mathcal{C}(x_i)$ containing all distinct haplotypes carrying the core haplotype, $c \in \mathcal{C}$, at x_0 and extending to marker x_i. Note that

$$\mathcal{C}(x_i) = \bigcup_{c \in \mathcal{C}} \mathcal{H}_c(x_i). \tag{2}$$

Following the example above, if the derived allele (1) is chosen as the core haplotype, then $\mathcal{H}_1(x_1) := \{11, 10\}$. Similarly, if the ancestral allele is the core haplotype, then $\mathcal{H}_0(x_1) := \{00, 01\}$

We calculate the EHH of the chromosomes carrying the core haplotype c to marker x_i as

$$EHH_c(x_i) = \sum_{h \in \mathcal{H}_c(x_i)} \frac{\binom{n_h}{2}}{\binom{n_c}{2}}, \tag{3}$$

where n_h is the number of observed haplotypes of type $h \in \mathcal{H}_c(x_i)$ and n_c is the number of observed haplotypes carrying the core haplotype ($c \in \mathcal{C}$).

4.2 Integrated Haplotype Score (iHS)

Integrated Haplotype Score (iHS) was introduced by Voight *et al.* (2006). iHS is calculated by using Equation 3 to track the decay of haplotype homozygosity for both the ancestral and derived haplotypes extending from a query site. To calculate iHS at a site, we first calculate the integrated haplotype homozygosity (iHH) for the ancestral (0) and derived (1) haplotypes ($\mathcal{C} := \{0, 1\}$) via trapezoidal quadrature.

$$\begin{aligned} iHH_c = & \\ & \sum_{i=1}^{|\mathcal{D}|} \frac{1}{2} \left(EHH_c(x_{i-1}) + EHH_c(x_i) \right) g(x_{i-1}, x_i) + \\ & \sum_{i=1}^{|\mathcal{U}|} \frac{1}{2} \left(EHH_c(x_{i-1}) + EHH_c(x_i) \right) g(x_{i-1}, x_i), \end{aligned} \tag{4}$$

where $\mathcal{D}$ is the set of markers downstream from the current locus such that $x_i \in \mathcal{D}$ denotes the i^{th} closest downstream marker from the locus of interest (x_0). $\mathcal{U}$ and $x_i \in \mathcal{U}$ are defined similarly for upstream markers. $g(x_{i-1}, x_i)$ gives the genetic distance between two markers. The (unstandardized) iHS is then calculated as

$$\ln\left(\frac{iHH_1}{iHH_0}\right). \tag{5}$$

Note that this definition differs slightly from that in Voight *et al.* (2006), where unstandardized iHS is defined with iHH_1 and iHH_0 swapped.

Finally, the unstandardized scores are normalized in frequency bins across the entire genome.

$$iHS = \frac{\ln\left(\frac{iHH_1}{iHH_0}\right) - E_p\left[\ln\left(\frac{iHH_1}{iHH_0}\right)\right]}{SD_p\left[\ln\left(\frac{iHH_1}{iHH_0}\right)\right]}, \tag{6}$$

where $E_p\left[\ln\left(\frac{iHH_1}{iHH_0}\right)\right]$ and $SD_p\left[\ln\left(\frac{iHH_1}{iHH_0}\right)\right]$ are the expectation and standard deviation in frequency bin p.

In practice, the summations in Equation 4 are truncated once $EHH_c(x_i) < 0.05$ or the computation extends more than 1Mbp from the core. Additionally with low density SNP data, if the physical distance b (in kbp) between two markers is > 20, then $g(x_{i-1}, x_i)$ is scaled by a factor of $20/b$ in order to reduce possible spurious signals induced by lengthy gaps. During computation if the start/end of a chromosome arm is reached before $EHH_c(x_i) < 0.05$ or if a gap of $b > 200$ is encountered, the iHS calculation is aborted for that locus. iHS is not reported at core sites with minor allele frequency < 0.05. In `selscan`, the EHH truncation value, gap scaling factor, and core site MAF cutoff value are all flexible parameters definable on the command line.

4.3 Cross-population Extended Haplotype Homozygosity (XP-EHH)

Cross-population Extended Haplotype Homozygosity (XP-EHH) was introduced by Sabeti *et al.* (2007). To calculate XPEHH between populations A and B at a marker x_0, we first calculate iHH for each population separately, integrating the EHH of the entire sample in the population (Equation 1).

$$\begin{aligned} iHH = & \\ & \sum_{i=1}^{|\mathcal{D}|} \frac{1}{2}\left(EHH(x_{i-1}) + EHH(x_i)\right) g(x_{i-1}, x_i) + \\ & \sum_{i=1}^{|\mathcal{U}|} \frac{1}{2}\left(EHH(x_{i-1}) + EHH(x_i)\right) g(x_{i-1}, x_i) \end{aligned} \tag{7}$$

If iHH_A and iHH_B are the iHHs for populations A and B, then the (unstandardized) XPEHH is

$$\ln\left(\frac{iHH_A}{iHH_B}\right), \tag{8}$$

and after genome-wide normalization we have

$$XPEHH = \frac{\ln\left(\frac{iHH_A}{iHH_B}\right) - E\left[\ln\left(\frac{iHH_A}{iHH_B}\right)\right]}{SD\left[\ln\left(\frac{iHH_A}{iHH_B}\right)\right]}. \tag{9}$$

In practice, the sums in each of iHH_A and iHH_B (Equation 7) are truncated at x_i—the marker at which the EHH of the haplotypes *pooled across populations* is $EHH(x_i) < 0.05$ or if the computation extends more than 1Mbp from the core. Scaling of $g(x_{i-1}, x_i)$ and handling of gaps is done as for iHS, and these parameters are definable on the `selscan` command line.

4.4 nSL

nSL is a statistic related to iHS and was introduced by Ferrer-Admetlla *et al.* (2014). nSL can be reformulated to conform to the notation given above. nSL is calculated as a log-ratio of the SL statistic calculated for the ancestral and derived haplotype pools. The essential difference from iHS is the removal of genetic map information. Whereas in Equation 4 we integrate with respect to a genetic map, for the calculation of nSL $g(x_i, x_j) = |j - i|$. Thus,

$$\begin{aligned} SL_c = & \\ & \sum_{i=1}^{|\mathcal{D}|} \frac{1}{2} \left(EHH_c(x_{i-1}) + EHH_c(x_i) \right) g(x_{i-1}, x_i) + \\ & \sum_{i=1}^{|\mathcal{U}|} \frac{1}{2} \left(EHH_c(x_{i-1}) + EHH_c(x_i) \right) g(x_{i-1}, x_i), \end{aligned} \tag{10}$$

and

$$nSL = \frac{\ln\left(\frac{SL_1}{SL_0}\right) - E_p\left[\ln\left(\frac{SL_1}{SL_0}\right)\right]}{SD_p\left[\ln\left(\frac{SL_1}{SL_0}\right)\right]}. \tag{11}$$

Note that, for nSL, $g(x_{i-1}, x_i) = 1$.

For the nSL option, there is no EHH decay cutoff, but the computation stops when more than 200 snps are included in building the haplotypes (can be changed with `--max-extend-nsl`). Scaling of $g(x_{i-1}, x_i)$ and handling of gaps is done as for iHS, and these parameters are definable on the `selscan` command line.

4.5 Mean Pairwise Sequence Difference (π)

The mean pairwise sequence difference among a sample of n haplotypes is

$$\pi = \frac{1}{\binom{n}{2}} \sum_{i=1}^{n-1} i(n-i)\xi_i, \tag{12}$$

where ξ_i is the unfolded site frequency spectrum.

5 Program Options

Using the command line flag `--help`, will print a help dialog with a summary of each command line option.

5.1 Input Files

All genetic data is required to be coded 0/1 and must not contain missing data. Consecutive loci are assumed to be in order with respect to their physical location on the chromosome. `selscan` assumes only

one non-homologous chromosome per file, and different non-homologous chromosomes should be computed separately.

These methods (excluding π) assume phased haplotypes. If your haplotypes are unphased you will have substantially reduced power to identify regions undergoing positive selection. Two popular programs that perform haplotype phasing are SHAPEIT2 (Delaneau *et al.*, 2013) and Beagle v4.0 (Browning and Browning, 2007).

Any file format that `selscan` supports may be directly read as a gzipped (`http://www.gzip.org/`) version without first decompressing.

5.1.1 `--hap` and `--ref`

Use `--hap` to specify a .hap file containing genetic variant information. If computing XP-EHH, use `--ref` to specify a .hap formatted sample from the desired reference population. Reference population files are expected to contain the exact same loci as the non-reference file.

A .hap file organizes variant data with rows representing a single haploid copy from an individual and columns representing consecutive loci delimited by whitespace. For example,

```
0 1 0 0 1
1 1 0 0 0
0 1 1 1 0
0 0 0 0 1
```

represents four halpoid samples with variant information at five loci.

Note that `selscan` expects a .map file to provide genetic and physical map information.

5.1.2 `--vcf` and `--vcf-ref`

Use `--vcf` to specify a .vcf file (see `https://github.com/samtools/hts-specs` for exact specifications) containing genetic variant information. If computing XP-EHH, use `--vcf-ref` to specify a .vcf formatted sample from the desired reference population. Reference population files are expected to contain the exact same loci as the non-reference file.

A .vcf file organizes variant data with rows representing consecutive loci and columns delimited by whitespace representing a diploid sample. For a given diploid sample at a particular locus a genotype is represented by two alleles separated by either / or |. The first nine columns contain information about the locus and the file is organized in the following way:

```
<chr#> <physical position> <id> <reference allele> <alternate allele> <
   quality> <filter> <info> <format> <individual 1 genotype> ... <
   individual N genotype>
```

All rows preceeded by a # symbol will be ignored. While a VCF file can encode quite a lot of information, `selscan` assumes biallelic phased genetic data and will not perform any checks on those assumptions nor perform any filtering (except filtering low frequency variants if `--skip-low-freq` is set). In fact, a .vcf file passed to `selscan` need not strictly conform to the general specifications. For example,

```
1 100 rs1 0 1 . . . . 0|1 0|0
1 200 rs2 0 1 . . . . 1|1 1|0
1 300 rs3 0 1 . . . . 0|0 1|0
1 400 rs4 0 1 . . . . 0|0 1|0
1 500 rs5 0 1 . . . . 1|0 0|1
```

is sufficient to represent two dipoid samples with variant information at five loci.

Note that even though physical position information is given on column two, `selscan` expects a .map file to provide genetic and physical map information.

5.1.3 `--tped` and `--tped-ref`

Use `--tped` to specify a .tped (transposed PLINK; Purcell *et al.* (2007)) file (see `http://pngu.mgh.harvard.edu/` for exact specifications) containing genetic variant information. If computing XP-EHH, use `--tped-ref` to specify a .tped formatted sample from the desired reference population. Reference population files are expected to contain the exact same loci as the non-reference file. Note that `selscan` expects .tped formatted data to be coded 0/1 only.

A .tped file organizes variant data with rows representing consecutive loci and columns delimited by whitespace representing haploid samples. The first four columns contain information about the locus and the file is organized in the following way:

```
<chr#> <id> <genetic position> <physical position> <haploid copy 1> ... <
   haploid copy N>
```

For example,

```
1 rs1 0.01 100 0 0 1 0 0
1 rs2 0.02 200 0 1 1 1 0
1 rs3 0.03 300 0 0 0 1 0
1 rs4 0.04 400 0 0 0 1 0
1 rs5 0.05 500 0 1 0 0 1
```

is sufficient to represent two dipoid samples with variant information at five loci.

Note that for .tped files `selscan` does not expect a .map file to provide genetic and physical map information, as this information is contained in the first four columns of the file. When calculating XP-EHH, map information is taken only from the file specifed with `--tped`.

5.1.4 `--map`

For all file formats except .tped, `selscan` requires a PLINK (Purcell *et al.*, 2007) formatted map file

The columns are delimited by whitespace and contain information about each locus. The file is organized in the following way:

```
<chr#> <id> <genetic position> <physical position>
```

For example,

```
1 rs1 0.01 100
1 rs2 0.02 200
1 rs3 0.03 300
1 rs4 0.04 400
1 rs5 0.05 500
```

5.2 Output Files

`selscan` produces two files as output. Results are output to a .out file and a log is output to a .log file. The .log file will record the runtime parameters as well as any information regarding the exclusion of particular loci. Results (.out) files are formatted in the following way.

For EHH the file will be named `<outfile>.ehh.<locusID>[.alt].out` and formatted as

<physicalPos> <geneticPos> <'1' EHH> <'0' EHH>

For iHS the file will be named <outfile>.ihs[.alt].out and formatted as

<locusID> <physicalPos> <'1' freq> <ihh1> <ihh0> <unstandardized iHS>

For XP-EHH the file will be named <outfile>.xpehh[.alt].out and formatted as

<locusID> <physicalPos> <geneticPos> <popA '1' freq> <ihhA> <popB '1' freq> <ihhB> <unstandardized XPEHH>

For nSL the file will be named <outfile>.nsl[.alt].out and formatted as

<locusID> <physicalPos> <'1' freq> <sl1> <sl0> <unstandardized nSL>

For π the file will be named <outfile>.pi.<winsize>bp.out and formatted as

<window start> <window end> <pi>

5.2.1 --out

Use --out to provide a base name for an output file. This will be used in place of <outfile> above. Default value is outfile.

5.3 Choice of Statistic

5.3.1 --ehh <string>

Use --ehh <Locus ID> to specify a core locus and calculate EHH curves for the ancestral and derived haploytpes extending out to a fixed distance. Associated flags: --ehh-win.

5.3.2 --ihs

Set --ihs to calculate iHS (see Section 4.2). Associated flags: --cutoff, --maf, --gap-scale, --max-gap, --max-extend, --trunc-ok, --alt, --skip-low-freq.

5.3.3 --xpehh

Set --xpehh to calculate XP-EHH (see Section 4.3). Associated flags: --cutoff, --gap-scale, --max-gap, --max-extend, --trunc-ok, --alt.

5.3.4 --nsl

Set --ihs to calculate iHS (see Section 4.2). Associated flags: --maf, --gap-scale, --max-gap, --max-extend-ns --trunc-ok, --alt.

5.3.5 --pi

Set --pi to calculate the mean pairwise sequence difference within a sliding window along the genome (see Section 4.5). Associated flag: –pi-win.

5.4 Controling How a Statistic is Computed

5.4.1 `--cutoff <double>`

Use `--cutoff` to set the EHH decay stopping condition. When computing iHS or XP-EHH, the EHH decay curve is truncated and integrated once the EHH decay cutoff is reached. Default is 0.05.

5.4.2 `--maf <double>`

Use `--maf <double>` to set a minor allele frequency (MAF) threshold. Any site below this will not be used as a core site for iHS and nSL scans. Default is 0.05.

5.4.3 `--gap-scale <int>`

Use `--gap-scale <int>` to set the gap scale parameter (Voight *et al.*, 2006). When computing iHS, XP-EHH, and nSL, if a gap of B bp is encountered and is greater than GAP_SCALE, then the distance function $g(x_i, x_j)$ is weighted by GAP_SCALE/B. Default is $20,000$ bp.

5.4.4 `--max-gap <int>`

Use `--max-gap <int>` to set the maximum allowed gap between loci when assembling haplotypes for iHS, XP-EHH, and nSL computations. If a gap greater than this is encountered before a stop condition is reached, the computation at the current core locus is aborted. Default is $200,000$ bp.

5.4.5 `--max-extend <int>`

Use `--max-extend <int>` to set an additional stopping condition for iHS and XP-EHH computations. If the EHH decay curve has extended MAX_EXTEND bp away from the core without reaching the ehh decay cutoff, truncate the curve here and integrate. Default is $1,000,000$ bp; set ≤ 0 for no restriction.

5.4.6 `--max-extend-nsl <int>`

Use `--max-extend-nsl <int>` to set a stopping condition for nSL computations. If the EHH decay curve has extended MAX_EXTEND loci away from the core, truncate the curve here and integrate. Default is 100 loci; set ≤ 0 for no restriction.

5.4.7 `--trunc-ok`

Core loci near the boundaries of the data set are unlikely to reach a stopping condition before running out of haplotype information. Typically in this case, EHH curves are truncated and thrown out. Set `--trunc-ok` to integrate these anyway.

5.4.8 `--alt`

Set `--alt` to compute EHH using sample haplotype frequencies:

$$EHH(x_i) = \sum_{h \in \mathcal{C}(x_i)} \left(\frac{n_h}{n}\right)^2 \tag{13}$$

and

$$EHH_c(x_i) = \sum_{h \in \mathcal{H}_c(x_i)} \left(\frac{n_h}{n_c}\right)^2. \tag{14}$$

Contrast with Equations 1 and 3.

5.4.9 --skip-low-freq

Set `--skip-low-freq` during an iHS scan to pre-filter all sites with a MAF less than that specified with `--maf`. If sites are not pre-filtered, `selscan` will use these sites to construct haplotypes, but will not use them as core sites (and thus will not report a score).

5.4.10 --ehh-win <int>

Set `--ehh-win <int>` to define for a single EHH computation the maximum extension in base pairs from the query locus. Default is $100,000$ bp.

5.4.11 --pi-win <int>

Set `--pi-win <int>` to define the size of the non-overlapping windows in base pairs for calculating π. Default is 100 bp.

5.5 Other Options

5.5.1 --threads <int>

`selscan` uses shared memory parallelism to speed up computations on multi-core workstations. Use `--threads <int>` to set the number of concurrent threads that `selscan` will use. Because genomic regions may have varying levels of haplotype homozygosity, slower EHH decays, and thus longer compute times, `selscan` assigns consecutive loci to different threads instead of partitioning the genome into chunks. Default is 1.

6 Change Log

```
07MAY2015 - Release of 1.1.0. Added option to calculate the nSL statistic
    described in A Ferrer-Admetlla, et al. (2014) MBE, 31: 1275-1291.
   Also introduced a check on map distance ordering.  Two new command
   line options.

--nsl <bool>: Set this flag to calculate nSL.
        Default: false

--max-extend-nsl <int>: The maximum distance an nSL haplotype is allowed
   to extend from the core.
        Set <= 0 for no restriction.
        Default: 100

06MAY2015 - Release of 1.0.5. Added basic VCF support.  selscan can now
   read .vcf and .vcf.gz files but without tabix support.  A mapfile is
   required when using VCF.  Two new command line options.

--vcf <string>: A VCF file containing haplotype data.
        A map file must be specified with --map.
```

--vcf-ref <string>: A VCF file containing haplotype and map data.
        Variants should be coded 0/1. This is the 'reference'
        population for XP-EHH calculations and should contain the same
           number
        of loci as the query population. Ignored otherwise.

07JAN2015 - norm bug fix and --skip-low-freq works for single EHH queries
   .

12NOV2014 - The program norm has been updated to allow for user defined
   critical values. Two new command line options.

--crit-percent <double>: Set the critical value such that a SNP with iHS
   in the most extreme CRIT_PERCENT tails (two-tailed) is marked as an
   extreme SNP.
        Not used by default.

--crit-val <double>: Set the critical value such that a SNP with |iHS| >
   CRIT_VAL is marked as an extreme SNP. Default as in Voight et al.
        Default: 2.00

17OCT2014 - Release of 1.0.4. A pairwise sequence difference module has
   been introduced. This module isn't multithreaded at the moment, but
   still runs quite fast. Calculating pi in 100bp windows with 198
   haplotypes with 707,980 variants on human chr22 finishes in 77s on the
    test machine. Using 100kb windows, it finishes in 34s. Two new
   command line options.

--pi <bool>: Set this flag to calculate mean pairwise sequence difference
    in a sliding window.
        Default: false

--pi-win <int>: Sliding window size in bp for calculating pi.
        Default: 100

15SEP2014 - Release of 1.0.3. **A critical bug in the XP-EHH module was
   introduced in version 1.0.2 and had been fixed in 1.0.3. Do not use
   1.0.2 for calculating XP-EHH scores.** Thanks to David McWilliams for
    finding this error. 1.0.3 also introduces support for gzipped input
   files. You may pass hap.gz, map.gz. and tped.gz files interchangably
   with unzipped files using the same command line arguments. A new
   command line option is available.

--trunc-ok <bool>: If an EHH decay reaches the end of a sequence before
   reaching the cutoff,
        integrate the curve anyway (iHS and XPEHH only).
        Normal function is to disregard the score for that core.

```
        Default: false

17JUN2014 - Release of 1.0.2.  General speed improvements have been made,
    especially with threading.  New support for TPED formatted data and
   new command line options are available.

--skip-low-freq <bool>: Do not include low frequency variants in the
   construction of haplotypes (iHS only).
        Default: false

--max-extend: The maximum distance an EHH decay curve is allowed to
   extend from the core.
        Set <= 0 for no restriction.
        Default: 1000000

--tped <string>: A TPED file containing haplotype and map data.
        Variants should be coded 0/1
        Default: __hapfile1

--tped-ref <string>: A TPED file containing haplotype and map data.
        Variants should be coded 0/1. This is the 'reference'
        population for XP-EHH calculations and should contain the same
           number
        of loci as the query population. Ignored otherwise.
        Default: __hapfile2

10APR2014 - Release of 1.0.1.  Minor bug fixes. XP-EHH output header is
   now separated by tabs instead of spaces.  Removed references to
   missing data (which is not accepted), and introduced error checking in
    the event of non-0/1 data being provided.

26MAR2014 - Initial release of selscan 1.0.0.
```

References

Browning, S. R. and Browning, B. L. 2007. Rapid and accurate haplotype phasing and missing data inference for whole genome association studies by use of localized haplotype clustering. *American Journal of Human Genetics*, 81: 1084–1097.

Delaneau, O., Zagury, J. F., and Marchini, J. 2013. Improved whole chromosome phasing for disease and population genetic studies. *Nature Methods*, 10: 5–6.

Ferrer-Admetlla, A., Liang, M., Korneliussen, T., and Nielsen, R. 2014. On detecting incomplete soft or hard selective sweeps using haplotype structure. *Molecular Biology and Evolution*, 31: 1275–1291.

Purcell, S., Neale, B., K., T.-B., Thomas, L., Ferreira, M. A. R., Bender, D., Maller, J., Sklar, P., de Bakker, P. I. W., Daly, M. J., and Sham, P. C. 2007. PLINK: a toolset for whole-genome association and population-based linkage analysis. *American Journal of Human Genetics*, 81: 559–575.

Sabeti, P. C., Reich, D. E., Higgins, J. M., Levine, H. Z. P., Richter, D. J., Schaffner, S. F., Gabriel,

S. B., Platko, J. V., Patterson, N. J., McDonald, G. J., Ackerman, H. C., Campbell, S. J., Altshuler, D., Cooper, R., Kwiatkowski, D., Ward, R., and Lander, E. S. 2002. Detecting recent positive selection in the human genome from haplotype structure. *Nature*, 419: 832–837.

Sabeti, P. C., Varilly, P., Fry, B., Lohmueller, J., Hostetter, E., Cotsapas, C., Xie, X., Byrne, E. H., McCarroll, S. A., Gaudet, R., Schaffner, S. F., and Lander, E. S. 2007. Genome-wide detection and characterization of positive selection in human populations. *Nature*, 449(7164): 913–918.

Szpiech, Z. A. and Hernandez, R. D. 2014. selscan: an efficient multithreaded program to perform EHH-based scans for positive selection. *Molecular Biology and Evolution*, 31: 2824–2827.

Voight, B. F., Kudaravalli, S., Wen, X., and Pritchard, J. K. 2006. A map of recent positive selection in the human genome. *PLoS Biology*, 4: e72.