Documentation for Software: *selink* v1.0

Guillaume Laval Pierre Boutillier Etienne Patin

Software from:

http://github.com/h-e-g/selink

December 8, 2017

Contents

1. Introduction 3
2. Getting started 3
   2.1. Compilation 3
   2.2. Quick Start 3
3. Input Files 4
   3.1. **.selink.hap* File 4
   3.2. **.selink.sample* File 4
   3.3. **.selink.legend* File 5
4. Generating Input Files From SHAPEIT2 Outputs 5
   4.1. *Extract_ancestral.pl* 5
   4.2. *Add_population.pl* 6
5. Running *selink* 6
   5.1. Options: Window Parameters 7
   5.2. Options: Neutrality Statistics 7
   5.3. Filtering populations and SNPs 8
6. Output Files 8
   6.1. **<pop1>.out* File 9
   6.2. **<pop1>-<pop2>.out* File 9
   6.3. **<pop1>_excluded.out* File 9
7. Normalization 9
8. Examples of Usage 9
9. List of Options 9

1. Introduction

Recent advances in large-scale sequencing are driving a wave of studies that aim to uncover, at an unprecedented level, variants that underlie disease risk, quantitative traits and genetic adaptation in several model species. In this context, genome-wide association studies could be systematically complemented with scans of signals of recent positive selection. Molecular signatures of selection, detected with neutrality statistics, may help understanding the evolutionary history of diseases and traits, including the high prevalence of disease resistance and susceptibility in certain species and high population variation in a number of phenotypes. We have developed the *selink* software in an attempt to make neutrality statistics computable on datasets made of thousands of individuals and millions of SNPs, in a limited amount of time. We hope that this effort will facilitate research in the fields of evolutionary genomics, epidemiology genomics, quantitative genomics, molecular ecology and evolutionary medicine.

Using a flexible **sliding-window** approach over entire chromosomes, *selink* computes neutrality statistics based (i) on extended haplotype homozygosity, including iHS, DIND, ΔiHH and XP-EHH; (ii) on population differentiation, including AMOVA-based pairwise F_{ST}, global F_{ST}, ΔDAF and PBS; and (iii) on the site frequency spectrum (SFS), including Tajima's *D*, Fu & Li's *F*, Fu & Li's *D* and Fay & Wu's H_n (using the general framework developed by Guillaume Achaz). The program takes as inputs **phased** individual haplotypes, SNP positions with **ancestral** and **derived** alleles, as well as individuals' **populations**. These files can be obtained automatically with perl scripts accompanying *selink* from the outputs of the haplotype phasing software SHAPEIT v.2, which handles both bed/bim/fam PLINK and VCF files.

2. Getting started

2.1. Compilation

The *selink* software can be downloaded from the repository website http://github.com/h-e-g/selink. The archive file includes the sources of the program coded in C, the documentation, perl scripts that help in formatting input files and in analyzing output files, as well as files used to compile the program. In order to compile *selink* on your machine, you must use the following commands:

```
> cd selink/
> aclocal
> autoheader
> autoconf
> automake --add-missing
> ./configure
> make
```

Optionally, you can check that compilation worked with this command:

```
> make check
```

2.2. Quick Start

After compilation, you first have to generate *selink* input files. Phasing the data is a prerequisite. We provide perl scripts to obtain *selink* input files from the outputs of the phasing program SHAPEIT v.2 (http://shapeit.fr/). The perl script *Extract_ancestral.pl* converts SHAPEIT2 *.*haps* output file into *.*selink.hap* and *.*selink.legend* files. The former extracts the ancestral state of each SNP, using *human_ancestor.fa.gz* annotation files (provided for the human species only).

> perl Extract_ancestral.pl -h <*.*haps file*> -o <*path/prefix of output files*> -f 1 -u 1

The perl script *Add_population.pl* generates *.*selink.sample* file, by adding to the *.*sample* file a required column of population labels.

> perl Add_population.pl -s <*.*sample file*> -f <*population file*> -o <*path/prefix of output files*>

The *selink* program can then be run on the three newly-generated files: *.*selink.hap,* *.*selink.legend* and *.*selink.sample*. Here, *selink* will compute iHS and XP-EHH using sliding windows of 200kb.

> ./selink -l 200000 -s -i -o <*prefix of output files*> <*prefix of input files*>

We invite you to use (and adapt) the shell script *submit_selink.sh*, developed to submit and parallelize file preparation and *selink* calculations by chromosome on a cluster of computers.

3. Input Files

3.1. *.*selink.hap* File

The *.*selink.hap* file contains the phased haplotypes of individuals. By default, each row is a SNP, and each column is made of the allelic states of a given haplotype. The file thus consists of *M* rows, *M* being the number of SNPs, and *pn* columns, *p* being the ploidy and *n* the sample size. *Selink* accepts that columns (haplotypes) have no separator, or are separated by a white space. Alleles are represented by 0 and 1, corresponding to *a0* and *a1* in the *.*selink.legend* file, as generated by SHAPEIT v.2.

SNP 1	0 0 0 0 0 0 0 0 0 0		0000000000
SNP 2	0 0 1 1 0 0 0 0 1 0		0011000010
SNP 3	0 0 0 0 0 0 0 0 0 0		0000000000
SNP 4	0 0 0 0 1 0 0 0 0 1		0000100001
SNP 5	0 0 0 0 0 0 0 0 0 0		0000000000
SNP 6	0 0 0 0 0 0 0 0 0 0	or	0000000000
SNP 7	0 0 0 0 1 0 0 0 0 1		0000100001
SNP 8	0 0 0 0 1 0 0 0 0 1		0000100001
SNP 9	0 0 0 0 0 0 0 0 0 0		0000000000
SNP 10	0 0 0 0 0 0 0 0 0 0		0000000000
SNP 11	0 1 0 0 0 0 0 1 0 0		0100000100

3.2. *.*selink.sample* File

The *.*selink.sample* file contains the list of individuals, in the same order as haplotypes in the *.*selink.hap* file. Each row is an individual. Column 1 is the individual ID, column 2 the gender (1=male, 2=female; note that this column is currently not taken into account), and column 3 is the population label.

```
HG00096 1 GBR
HG00097 2 GBR
HG00099 2 GBR
HG00100 2 GBR
HG00101 1 GBR
NA20816 1 TSI
NA20818 2 TSI
NA20819 2 TSI
NA20826 2 TSI
NA20828 2 TSI
```

3.3. *.*selink.legend* File

The *.*selink.legend* file contains the list of SNPs, in the same order as SNPs in the *.*selink.hap* file. Each row is a SNP. Column 1 is the SNP identifier, column 2 is the physical or genetic position, column 3 is the allele *a0* (coded by a 0 in the *.*selink.hap* file), column 4 is the allele *a1* (coded by a 1 in the *.*selink.hap* file) and column 5 is the ancestral state. This file should have a header.

```
ID position allele0 allele1 ancestral
rs149201999 16050408 T C T
rs146752890 16050612 C G C
rs139377059 16050678 C T c
rs188945759 16050984 C G G
rs6518357 16051107 C A C
rs62224609 16051249 T C T
rs62224610 16051347 G C G
rs143503259 16051453 A C N
rs192339082 16051477 C A c
```

4. Generating Input Files From SHAPEIT2 Outputs

In order to generate the above-mentioned input files and run *selink*, you need to add some extra information to the outputs of SHAPEIT v.2.

4.1. *Extract_ancestral.pl*

Statistics based on extended haplotype homozygosity like iHS require the ancestral state of each SNP. To obtain ancestral states, we provide a perl script, *Extract_ancestral.pl*, that reads the *.*haps* SHAPEIT v.2 output, extracts physical position of each SNP, searches for the ancestral state at this position in *human_ancestor.fa.gz* annotation files (provided for the human species only), checks if this ancestral state

matches with the *a0* or *a1* alleles and eventually reports the proportions of SNPs whose ancestral state matches *a0*, *a1*, none of them, or is not found in the annotation files.

> perl Extract_ancestral.pl -h <**.haps file*> -o <*path/prefix of output files*> -f 1 -u 1

There are two options:

-f, which inactivates (*-f 0*, the default) or activates (*-f 1*) the mode that automatically flips *a0* and *a1* alleles when none of them matches the extracted ancestral state. This option is *a priori* useless if you analyse VCF files, but is highly recommended if you analyse SNP array data in which alleles have not yet been matched to the forward strand. *Extract_ancestral.pl* expects *human_ancestor.fa.gz* annotation files in the directory where the perl script is executed (but this can be easily changed in the script, if necessary).

-u, which inactivates (*-u 0*, the default) or activates (*-u 1*) the mode that considers uncertain ancestral states (in lower case in *human_ancestor.fa.gz* annotation files, see human_ancestor_GRCh37_e59.README) as actual ancestral states (in upper case in *human_ancestor.fa.gz* annotation files).

4.2. *Add_population.pl*

Statistics based on population differentiation like F_{ST} require that the sample is divided into populations. This information should be provided in the third column of **.selink.sample*. Of note, if you have a unique population and want to estimate only intrapopulation statistics, you still have to fill this column with a population label, identical for all individuals. We provide a perl script, *Add_population.pl*, that reads the **.sample* SHAPEIT v.2 output and a population file, and generates the **.selink.sample* file by simply matching individuals in both files and reporting the population labels found in the population file.

> perl Add_population.pl -s <**.sample file*> -f <*population file*> -o <*path/prefix of output files*> -c 3

The population file is a list of individuals that should include all individuals in the **.sample* SHAPEIT v.2 output. Each row is an individual. Column 1 must be the individual ID. The column reporting the population label is column 2 by default, but can be specified with the *–c* option (column 3 in the example above). The order of individuals can be different from the one in the **.sample* SHAPEIT v.2 output. There can be more individuals than in the **.sample* SHAPEIT v.2 output.

5. Running *selink*

The basic command line of *selink* is as follows:

> ./selink *-options* <*prefix of input files*>

The default path and prefix of output files are those of input files. They can nevertheless be changed with the *–o* option. Several additional options, listed below, can be used to specify the parameters of sliding windows and which neutrality statistics to compute.

5.1. Options: Window Parameters

Selink has been designed to estimate neutrality statistics along entire chromosomes, using a sliding-window approach. Each window is defined by a core SNP, and the pace of the sliding approach is always of one SNP, so that there is one row in the output files for each SNP present in the input files. The size of the window can be modified with options *–n*, *–l* or *-P*. With the option *–n <number of SNPs>*, windows will be defined based on a number of SNPs surrounding the core SNP that defines each window. With the option *–l <size in bp or cM>*, windows will be defined based on physical or genetic distances, depending on the nature of positions reported in the **.selink.legend* file. When using the *–l* option, windows are bounded by the SNPs that are the closest to the limit defined by the bp or cM window size. In some applications, you may want to use manually-defined windows. This can be done with the option *–P <position file>*. The position file should list on each row the core SNP position and the positions of the window boundaries around this core SNP.

SNPs within each sliding window are used differently, depending on which neutrality statistics are calculated. Extended haplotype statistics (iHS, DIND, ΔiHH and XP-EHH) are SNP-centric, and are thus calculated for the core SNP, using all SNPs included in the window. Of note, the option *–S* allows to estimate iHS, ΔiHH and XP-EHH outside of window limits, and uses the EHH<0.05 criterion instead (see details below). Site frequency spectrum (SFS)-based statistics (Tajima's *D*, Fu & Li's *F*, Fu & Li's *D* and Fay & Wu's H_n) are calculated on all SNPs included in the window. Finally, population differentiation statistics (AMOVA-based F_{ST}) are SNP-centric, and are computed only for the core SNP (i.e., the other SNPs in the window are not considered).

5.2. Options: Neutrality Statistics

Five options in *selink* allow to compute different sets of neutrality statistics: intra-population statistics based on extended haplotype homozygosity (iHS) with option *–s*; intra-population statistics based on intra-allelic diversity (DIND) with option *–p*; interpopulation statistics based on extended haplotype homozygosity (ΔiHH and XP-EHH) with option *–i*; interpopulation statistics based on population differentiation (AMOVA-based F_{ST}) with option *–i*; intrapopulation classical neutrality statistics based on the site frequency spectrum (Tajima's *D*, Fu & Li's *F*, Fu & Li's *D* and Fay & Wu's H_n) with option *–w*; intrapopulation custom statistics based on the site frequency spectrum (T_Ω test; Achaz G, Genetics 2009) with option *–c*.

Selink computes iHS as follows. At each core SNP, haplotypes carrying the ancestral and derived alleles are considered separately. Let's first consider haplotypes carrying the derived allele *D*, whose absolute frequency is C_D in the sample. We count the absolute frequency C_h of all n_i possible haplotypes observed in the data, with $h \in [1, n_i]$. Thus, $C_D = \sum_{h=1}^{n_i} C_h$. Haplotypes are defined as the observed combinations of alleles of a given set of SNPs, which include the core SNP and all SNPs until SNP *i*, SNP *i*=0 being the most proximal SNP to the core SNP. We then compute EHH_{iD}, the Extended Haplotype Homozygosity of the derived allele of the core SNP, measured at SNP *i*, as follows:

$$EHH_{iD} = \frac{\sum_{h=1}^{n_i} \binom{C_h}{2}}{\binom{C_D}{2}}$$

EHH_{iD} is computed until EHH_{i-1} is lower than 0.05 or until the SNP i is outside the boundary of the core window. Of note, one can override the latter limitation by using option *–X*, so that EHH_i is always computed until EHH_{i-1} is lower than 0.05.

Selink then computes iHH_D, the area under *EHH* curves in 5' and 3' of the core SNP. Classical algorithms compute C_h at each core SNP from scratch, by iteratively including surrounding SNPs, which makes computation time quadratic with the number of SNPs. With *selink*, we simply compute C_h at a core SNP by summing C_h values obtained for the previous core SNP, reducing drastically the complexity of the algorithm. The iHH_A value (for the ancestral allele) is obtained the same way. Finally, iHS is computed as follows:

$$iHS = \ln \frac{iHH_A}{iHH_D}$$

SFS-based neutrality statistics such as Tajima's *D*, Fu & Li's *F*, Fu & Li's *D* and Fay & Wu's H_n are obtained using the general framework proposed by Guillaume Achaz (Achaz G, *Genetics* 2009). Let's consider a genomic window that includes S SNPs obtained in n individuals. To obtain the site frequency spectrum, one counts the number of SNPs ξ_i with a derived allele absolute frequency of i in the sample, with $i \leq 2n$ and $\sum_i \xi_i = S$. It has been shown that classical estimators of the neutral parameter θ, such as θ_π used in Tajima's *D* statistics, can be expressed as the sum of $i\xi_i$ weighted by a given vector ω (Achaz G, *Genetics* 2009). Thus, classical neutrality statistics, which are based on the comparison of two estimators of θ normalized by its standard deviation, depends on the comparison of two weighting vectors.

$$T_\Omega = \frac{\sum_i \Omega_i i\xi_i}{\sqrt{\alpha_n \theta + \beta_n \theta^2}}$$

Where,

$$\Omega_i = \frac{\omega_{1i}}{\sum_i \omega_{1i}} - \frac{\omega_{2i}}{\sum_i \omega_{1i}}$$

Selink computes $i\xi_i$ in a given window, and then uses different weighting vectors to obtain neutrality statistics. For example, Tajima's *D* is obtained with $\omega_{1i} = n - 1$ and $\omega_{2i} = {}^1/_i$. A custom weighting vector can be specified with option *–c <file>*. This file lists $2n$ values on separate rows, which sum should equal 1. The T_Ω statistics will compare the custom weighting vector to Watterson's θ_S vector ($\omega_{2i} = {}^1/_i$).

5.3. Options: Filtering populations and SNPs

To be done

6. Output Files

The prefix of output files can be specified with *–o <prefix>* option. Three types of output files are generated, depending on the options used.

6.1. **<pop>.out* File

When intrapopulation statistics are computed (options *–s*, *–p* or *–w*), results are outputted to one **<pop>.out* file per analyzed population, where *pop* is the population name specified in the third column of the **.selink.sample* file. The file contains one row per SNP, which corresponds either to the core SNP for iHS or DIND statistics or the SNP central to the window for SFS-based neutrality statistics. Each row contains the core SNP identifier, its position, the 5' and 3' boundaries of the window around the SNP, within which all statistics are calculated, the core SNP derived allele frequency, and requested neutrality statistics.

6.2. **<pop1>-<pop2>.out* File

When interpopulation statistics are computed (options *–i* or *–w*), results are outputted to one **<pop1>-<pop2>.out* file per pair of analyzed population. The file is structured the same way as the intrapopulation file.

6.3. **<pop>_excluded.out* File

The <*pop*>_excluded.out file lists the SNPs that have been automatically excluded by *selink* from the list of core SNPs, because they were monomorphic or had no ancestral allele information.

7. Normalization

To be done

8. Examples of Usage

To be done

9. List of Options

Message options

-v, --version	Prints version number and exits.
-h, --help	Prints the list of options and exits.
-q, --quiet	Turns off stderr messages.

Input/output options

-o, --outfile <*prefix*>	Outputs results to <*prefix*>.* files. By default, results are outputted to stdout.
-f, --filter <*file*>	Keeps populations listed in <*file*>.
-P, --pos <*file*>	Keeps SNPs listed in <*file*>.

Window options

-l, --lenw	Window size defined in bases.
-n, --numb	Window size defined in number of SNPs

Statistics options

-s, --ihs	Computes iHS statistics.
-p, --pi	Computes DIND statistics.
-i, --inter	Computes XP-EHH and F_{ST} statistics.
-w, --omega	Computes Tajima's *D*, Fu & Li's *F*, Fu & Li's *D* and Fay & Wu's H_n statistics.
-c, --comp<*file*>	Computes T_Ω statistics based on the weighting vector in <*file*>.
-K, --freq	Computes haplotype diversity (the option substantially increases computation time).