Selection scans and downstream analysis with selscan

Amatur Rahman*[1ab], T. Quinn Smith†[1b], and Zachary A. Szpiech‡[1a]

[1]Department of Biology, The Pennsylvania State University, University Park, PA 16802, USA

[a]Correspondence: aur1111@psu.edu, zps5164@psu.edu

[b]Equal Contribution

Abstract

Summary statistics based on Extended Haplotype Homozygosity (EHH) are widely used for inferring positive selection in genomes as a result of their ease of use, computational efficiency, and interpretability. These various summary statistics can be applied to single populations or to pairs of populations, can be used with a genetic recombination map or without, and can be applied to phased or unphased data. Although they are straightforward to compute, there lacks clear descriptions on how they relate to one another, how they should be used, and how their results should be interpreted. Here, we provide a comprehensive introduction to selection statistics as they are implemented in the widely used software, `selscan`. We demonstrate the behavior of such statistics on simulated data and highlight best practices by performing an example downstream analysis on data from the 1000 Genomes Project using new features in `selscan v3.0`. We hope these guidelines will foster reproducibility in the evolutionary genomics community. Precompiled executables and source code for `selscan v3.0` can be found at `https://github.com/szpiech/selscan`.

*ORCID: 0000-0002-9166-1220
†ORCID: 0009-0004-9955-4915
‡ORCID: 0000-0001-6372-8224

1 Introduction

Among evolutionary biologists, there is great interest in identifying the genomic basis of adaptations. Pinpointing putatively adaptive mutations, especially in relation to known selective pressures and with functional validation, can provide insights into the biological mechanisms underlying phenotypic innovations. In humans, understanding the biological basis of adaptation can give a deeper understanding of our evolutionary history and can reveal important information about the relationship between genes, adaptation, and disease (Scheinfeldt and Tishkoff, 2013; Fu and Akey, 2013; Benton et al., 2021).

Given this importance, many statistics have been developed to make inferences about positive selection in the genome across different time scales (Booker et al., 2017; Pavlidis and Alachiotis, 2017). For detecting recent and strong positive selection, common methodologies typically make use of expected distortions in either the site frequency spectrum or haplotype patterns (Szpiech and Hernandez, 2016). When a strongly adaptive allele arises, it will sweep to high frequency on a timescale faster than recombination or mutation occur, bringing linked variation to high frequency as well. This results in a region of low genetic diversity, long haplotypes, and an excess of high and low frequency alleles. Although more complicated model-based (Nielsen et al., 2005; DeGiorgio et al., 2016; Harris and DeGiorgio, 2020; DeGiorgio and Szpiech, 2022; Stern et al., 2019; Vaughn and Nielsen, 2024; Hejase et al., 2022) and machine learning (Kern and Schrider, 2018; Sugden et al., 2018; Amin et al., 2024; Arnab et al., 2025) methods have proven useful, especially for inferring specific parameters of a sweep, summary-statistic-based methods remain widely used as a result of their ease of use, computational efficiency, and interpretability.

One particularly important class of summary statistics for inferring positive selection are those based on extended haplotype homozygosity (Sabeti et al., 2002), which are designed to capture the signal of long high frequency haplotypes in the vicinity of a sweep that is either ongoing or recently completed. EHH-based summary statistics include iHS (Voight et al., 2006), nSL (Ferrer-Admetlla et al., 2014), XP-EHH (Sabeti et al., 2007), and XP-nSL (Szpiech et al., 2021), and although originally formulated for use on phased haplotype data, they have also been extended to work on unphased data as well (Klassmann and Gautier, 2022; Szpiech, 2024). These statistics are also frequently included as important components in more advanced machine learning approaches

(Amin et al., 2024; Arnab et al., 2025). Given their broad adoption for use in selection inference analyses, they have been implemented in several widely used software programs including `selscan` (Szpiech and Hernandez, 2014; Szpiech, 2024; Rahman et al., 2025), rehh (Gautier and Vitalis, 2012; Gautier et al., 2017), and hapbin (Maclean et al., 2015). Currently, selscan is the leading implementation in terms of computational efficiency and memory usage (Rahman et al., 2025).

In this work, we introduce the definitions of these common EHH-based statistics and their interpretations. We also provide an overview of how to use `selscan v3.0` for genome-wide selection inference and downstream analysis.

2 Materials and Methods

First, we describe the main statistics (Sections 2.1, 2.2, 2.3 and 2.4) show the exact commands applied to simulated data are provided, matching those used to generate the results in the Results section, aiming to provide a quick start for users interested in selection scan analysis. We also highlight the importance of downstream analysis and present both its application and the relevant commands (Section 2.5). We describe our simulations and dataset preprocessing methods in Section 2.7.

2.1 EHH

2.1.1 Definition for phased data

Consider a sample of n chromosomes. Define $\mathcal{C}$ to be the set of all possible distinct haplotypes at a locus of interest, which is called x_0. In addition, define $\mathcal{C}(x_i)$ to denote the set of all possible distinct haplotypes extending from the locus x_0 to the ith marker either upstream or downstream from x_0. If x_0 biallelic locus, then let 0 represent the ancestral allele and 1 represent the derived allele. Therefore, $\mathcal{C}(x_1) := \{0, 1\}$. If x_1 is an immediately adjacent locus, then the set of all possible haplotypes becomes $\mathcal{C}(x_1) := \{11, 10, 00, 01\}$.

EHH between x_0 and x_1 of the entire sample (Sabeti et al., 2002; Szpiech and Hernandez, 2014) 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 $h \in \mathcal{C}(x_i)$.

It is advantageous to calculate the haplotype homozygosity of a sub-sample of chromosomes all carrying a 'core' allele at locus x_0. Define $\mathcal{H}_c(x_i)$ to be a partition of $\mathcal{C}(x_i)$ containing all unique haplotypes carrying the core haplotype, $c \in \mathcal{C}$, at x_0 and extending to marker x_i as

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

$\mathcal{H}_1(x_1) := \{11, 10\}$ and $\mathcal{H}_0(x_1) := \{00, 01\}$ when the core allele is chosen as the derived and ancestral allele at x_0, respectively.

Finally, define EHH of haplotypes containing the core allele, c, to a locus 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 chromosomes with haplotype $h \in \mathcal{H}_c(x_i)$ and n_c is the number of chromosomes carrying the core allele ($c \in \mathcal{C}$).

2.1.2 Definition for unphased data

EHH can be adapted to unphased genotypes of diploid individuals by encoding each locus with the number of observed derived alleles: 0, 1, 2 corresponding to homozygous ancestral, heterozygous, and homozygous derived, respectively. Let $\mathcal{C} := \{0, 1, 2\}$ denote the set of all possible genotypes at locus x_0. Define the set of all unique haplotypes extending from site x_0 to site x_i as $\mathcal{C}(x_i)$, where x_i is either upstream or downstream of x_0. For example, if x_1, the site adjacent to x_0, then $\mathcal{C}(x_1) := \{00, 01, 02, 10, 11, 12, 20, 21, 22\}$. With these updated definitions, EHH can be computed for a set of 'multi-locus genotypes' using Equation 1.

Similarly, the above reasoning can be applied when defining EHH for a 'core' allele. To compute the EHH of a subset of observed haplotypes that all contain the same 'core' haplotype, let $H_c(x_i)$ be the partition of $C(x_i)$ containing genotype $c \in C$ at x_0. For example, choosing the derived homozygous genotype ($c = 2$) as the core, $H_2 := \{20, 21, 22\}$. Thus, EHH can be computed for all individuals carrying a given genotype at site x_0 extending out to site x_i using Equation 3

Finally, define the complement EHH of a sample of haplotypes—the EHH of all haplotypes not

containing the core allele—as

$$cEHH_c(x_i) = \sum_{h \in \mathcal{H}_c(x_i) \setminus H_c(x_i)} \frac{\binom{n_h}{2}}{\binom{n_{c'}}{2}}, \tag{4}$$

where $n_{c'}$ is the number of observed haplotypes with a core genotype of not c.

2.1.3 Interpretation of EHH

EHH measures the decay of haplotype homozygosity in iteratively larger windows extending from a core locus. It is computed from among all haplotypes in the sample (Equation 1) or from among only haplotypes containing a particular core allele (Equation 3). Under neutrality, haplotype homozygosity is expected to decay quickly as the window grows larger—reflecting the underlying diversity of the haplotypes in the sample. However, haplotype homozygosity will be maintained at high levels in the vicinity of a recent or ongoing sweep. In this case, EHH of all samples (Equation 1) is expected to decay slowly as window size increases, as the sweeping haplotype will dominate the homozygosity calcualation. If computing EHH from among only haplotypes containing a particular core allele (Equation 3), the expected pattern depends on whether the core allele is (or is linked to) the adaptive allele. When the core allele is (or is linked to) the adaptive allele, EHH computed from among only the sweeping haplotypes will have a slow decay as window size increases. On the other hand, the EHH computed from among only non-sweeping haplotypes will have a fast decay as window size increases, similar to the expectation under neutrality. See Section 3.1.2 for an illustration.

2.1.4 Example Commands

To calculate EHH at a marker $rs1$ on phased data:

```
selscan --ehh rs1 --vcf input.vcf
```

Assuming the input VCF file contains one chromosome, we can calculate EHH at a specific position. For example, at 5MB:

```
selscan --ehh 5000000 --vcf input.vcf
```

Suppose we have a file with a list of three sites, *genetic.map*:

chr#	ID#	Genetic Position	Physical Position
1	rs1	0.09	1000010
1	rs2	0.10	1000013
1	rs3	0.12	2000000

Then, to run EHH at each of the three loci:

```
selscan --ehhs 1000010,1000013,2000000 --vcf input.vcf --map genetic.map
```

To run on unphased data at one locus, we add the `--unphased` flag.

```
selscan --ehh rs1 --vcf input.vcf --map genetic.map --unphased
```

2.2 iHS and nSL

2.2.1 Definition for phased data

The decay of haplotype homozygosity tracked by EHH and EHH_c provides useful biological information for identifying sweeps at individual loci of interest, but these statistics are not well-suited for genome-wide scans. This motivated the development of iHS, which uses Equation 3 to summarize the decay of ancestral and derived haplotype homozygosity into a single score. iHS is calculated at a site by first calculating the integrated haplotype homozygosity (iHH) for the ancestral (0) and derived (1) haplotypes ($\mathcal{C} := \{0, 1\}$) via trapezoidal quadrature.

$$
\begin{aligned}
iHH_c = & \\
& \sum_{i=1}^{|\mathcal{D}|} \frac{1}{2} \left(EHH_c(x_{i-1}) + EHH_c(x_i) \right) g(x_{i-1}, x_i) + \\
& \sum_{i=1}^{|\mathcal{U}|} \frac{1}{2} \left(EHH_c(x_{i-1}) + EHH_c(x_i) \right) g(x_{i-1}, x_i),
\end{aligned} \tag{5}
$$

$\mathcal{D}$ is the set of downstream loci, such that, if $x_i \in \mathcal{D}$, then x_i is the ith closest downstream locus from x_0. Similarly define $\mathcal{U}$ to be the set of upstream markers. Let $g(x_{i-1}, x_i)$ be the genetic or

physical distance between two loci. The unstandardized iHS is calculated as

$$rawIHS = \log\left(\frac{iHH_1}{iHH_0}\right). \tag{6}$$

Note, this definition is different from that in (Voight et al., 2006), where the roles of iHH_1 and iHH_0 are swapped and a natural logarithm is used.

A related statistic, nSL, was introduced by (Ferrer-Admetlla et al., 2014). Although Ferrer-Admetlla et al. define nSL in terms of the mean number of sites shared among all pairwise haplotypes in the vicinity of a query site, they also prove a reformulation in terms of haplotype homozygosity. Using Equation 5 above, the essential difference between nSL and iHS is that the distance function for nSL is given by $g(x_i, x_j) = |j - i|$.

2.2.2 Definition for unphased data

iHS and nSL can be adapted to unphased genotypes by using the definitions in Section 2.1.2 and defining the homozygous ancestral and the homozygous derived genotypes as $c = 0$ and $c = 2$, respectively. The integrated haplotype homozygosity (iHH) can be calculated for each genotype using Equation 5, and the complement integrated haplotype homozygosity (ciHH) can be calculated for both homozygous core genotypes as

$$ciHH_c = \sum_{i=1}^{|D|} \frac{1}{2}(cEHH_c(x_{i-1}) + cEHH_c(x_i) g(x_{i-1}, x_i) + \sum_{i=1}^{|U|} \frac{1}{2}(cEHH_c(x_{i-1}) + cEHH_c(x_i)) g(x_{i-1}, x_i), \tag{7}$$

where $g(x_{i-1}, x_i)$ is the genetic or physical distance between two loci.

The unstandardized unphased iHS is calculated as

$$rawIHS = \begin{cases} iHS_2 & \text{if } iHS_2 > iHS_0 \\ -iHS_0 & \text{if } otherwise \end{cases} \tag{8}$$

where $iHS_2 = log((iHH_2)/(ciHH_2))$ and $iHS_0 = log((iHH_0)/(ciHH_0))$. Unstandardized unphased nSL is computed similarly but with $g(x_i, x_j) = |j - i|$.

2.2.3 Interpretation of iHS and nSL

iHS/nSL are intended to efficiently contrast the decay of haplotype homozygosity computed from among haplotypes containing the ancestral and derived alleles at a locus of interest. From Equation 6, when the log-ratio takes an extreme positive value, this suggests haplotypes carrying the derived allele are unusually long and low-diversity compared to haplotypes carrying the ancestral allele. When the log-ratio takes an extreme negative value, this suggests haplotypes carrying the ancestral allele are unusually long and low-diversity compared to haplotypes carrying the derived allele. It is tempting, in this case, to think that only extreme positive values are of interest, assuming that an adaptive allele is necessarily derived. However, ancestral alleles in linkage disequilibrium with the adaptive allele also provide information in the vicinity of a sweep, and in some data sets the adaptive allele may not even be observed. Therefore analyses using iHS and nSL are typically concerned with extreme absolute scores. See Section 3.1.2 for an illustration.

2.2.4 Example Commands

To calculate iHS on phased data, `selscan` requires genetic data (e.g., VCF) and a genetic map file.

```
selscan --ihs --vcf input.vcf --map genetic.map
```

To calculate iHS on unphased data, add the `--unphased` flag.

```
selscan --ihs --vcf input.vcf --map genetic.map --unphased
```

To calculate nSL on phased data, `selscan` requires only genetic data (e.g., VCF). It uses physical postions included in it, specified via the flag `--pmap`.

```
selscan --nsl --vcf input.vcf --pmap
```

To calculate nSL on unphased data, add the `--unphased` flag.

```
selscan --nsl --vcf input.vcf --pmap --unphased
```

2.3 XPEHH and XPnSL

2.3.1 Definition for phased data

XPEHH (Sabeti et al., 2007) and XPnSL (Szpiech et al., 2021) were introduced as two-population extensions to iHS and nSL, respectively, intended to identify local adaptation. For these statistics, instead of comparing the EHH between haplotypes with the ancestral and derived alleles (i.e., by computing EHH_c for each allele type), they compare the EHH of all haplotypes in one population to another population. Consider two populations, A and B, and a locus x_0. First, iHH is calculated for each population by integrating the EHH of all samples (Equation 1) in each population.

$$\begin{aligned} iHH = & \\ & \sum_{i=1}^{|\mathcal{D}|} \frac{1}{2} \left(EHH(x_{i-1}) + EHH(x_i) \right) g(x_{i-1}, x_i) + \\ & \sum_{i=1}^{|\mathcal{U}|} \frac{1}{2} \left(EHH(x_{i-1}) + EHH(x_i) \right) g(x_{i-1}, x_i) \end{aligned} \tag{9}$$

Let iHH_A and iHH_B be the iHH for populations A and B, respectively. Then the unstandardized XPEHH is

$$rawXPEHH = \log \left(\frac{iHH_A}{iHH_B} \right), \tag{10}$$

XPnSL is calculated similarly, but with $g(x_{i-1}, x_i) = |j - i|$.

2.3.2 Definition for unphased data

Unphased XPEHH and XPnSL can be calculated using the unphased definitions given in Section 2.1.2 and Equation 9, which defines unphased iHH for the entire population. The distance measure is either centimorgans or base pairs for XPEHH (Sabeti et al., 2007), or the number of observed sites for XPnSL (Szpiech et al., 2021). Both XP statistics between population A and B are computed as

$$rawXP = \log \left(\frac{iHH_A}{iHH_B} \right) \tag{11}$$

2.3.3 Interpretation of XPEHH and XPnSL

XPEHH/XPnSL are intended to efficiently contrast the decay of haplotype homozygosity of a sample of haplotypes from one population to a sample of haplotypes from a closely related population at a locus of interest. From Equation 11, when the log-ratio takes an extreme positive value, this suggests haplotypes from the "A" population are unusually long and low-diversity compared to haplotypes from the "B" population. When the log-ratio takes an extreme negative value, this suggests haplotypes from the "B" population are unusually long and low-diversity compared to haplotypes from the "A". In this case, extreme positive values and extreme negative values are of interest, but they should be analyzed separately, as they suggest possible sweeps in one population or the other. See Section 3.1.3 for an illustration. Importantly, in the case where a sweep is occurring in both populations at the same locus, large values of iHH in the log-ratio would cancel out and the result is not distinguishable from neutrality.

2.3.4 Example Commands

To calculate XPEHH on phased data, `selscan` requires three input files, genetic data (e.g., VCF) for a reference population, genetic data for an alternate population, and a genetic map.

```
selscan --xpehh --vcf-ref ref.vcf --vcf alt.vcf --map genetic.map
```

To calculate XPEHH on on unphased data, add the `--unphased` flag.

```
selscan --xpehh --vcf-ref ref.vcf --vcf alt.vcf --unphased --map genetic.map
```

To calculate XPnSL on phased data, only two input files are required, genetic data (e.g., VCF) for a reference population, and enetic data for an alternate population.

```
selscan --xpnsl --vcf-ref ref.vcf --vcf alt.vcf
```

To calculate XPnSL on on unphased data, add the `--unphased` flag.

```
selscan --xpnsl --vcf-ref ref.vcf --vcf alt.vcf --unphased
```

2.4 Integrated Haplotype Homozygosity Pooled (iHH12)

The iHH12 (Torres et al., 2018) statistic is adapted from the H12 statistic (Garud et al., 2015) and is similar to iHH (Equation 9). However iHH12 integrates over EHH12, which is given by

$$EHH12(x_i) = \frac{\binom{n_{h_1}+n_{h_2}}{2}}{\binom{n}{2}} + \sum_{h \in \mathcal{C}(x_i) \setminus \{h_1, h_2\}} \frac{\binom{n_h}{2}}{\binom{n}{2}}, \tag{12}$$

where h_i is the ith most frequent haplotype in the sample and n_{h_i} is the number of observed h_i haplotypes. iHH12 is computed with Equation 9 but using $EHH12$ in place of EHH.

2.4.1 Interpretation of iHH12

iHH12 is intended to efficiently summarize the decay in haplotype homozygosity among all samples in a single population, with a particular emphasis on detecting soft sweeps. On the assumption that soft sweeps will likely have more than one haplotype sweeping to high frequency, iHH12 combines the counts of the two most frequent haplotypes into a single frequency class, thereby recovering homozygosity signal that would otherwise be lost by partitioning the homozygosity contribution between two classes. When iHH12 values are large, this suggests a collection of long haplotypes of low diversity, and may be indicative of a sweep in the vicinity of that locus. See Section 3.1.2 for an illustration.

2.4.2 Example Commands

To calculate iHH12 on phased data, `selscan` requires genetic data (e.g., VCF) and a genetic map file.

```
selscan --ihh12 --vcf input.vcf --map genetic.map
```

There is not currently an unphased version of iHH12.

2.5 Downstream analysis

`selscan`'s main functions compute unstandardized EHH-based statistics. However, normalization is necessary to draw meaningful conclusions, and other analyses, such as outlier detection and

gene-based scoring, are commonly performed but require special care in executing. `selscan v3.0` now includes a subcommand, `norm`, to facilitate downstream analysis of `selscan` results. In the following, we describe how `selscan`'s `norm` subcommand performs normalization, outlier detection, and gene scoring for potential selection signals.

2.5.1 Normalization

Normalization transforms the raw statistics to an approximate Standard Normal distribution, allowing them to be used for further statistical analysis and to be comparable between studies. For iHS and nSL (phased or unphased) this is a particularly important step, as the raw statistics are correlated with derived allele frequency, and therefore normalization is performed within frequency bins along the genome (see Section 3.1.1 for an illustration). These statistics are therefore normalized by

$$score = \frac{rawScore - E_p\Big[rawScore\Big]}{SD_p\Big[rawScore\Big]}, \tag{13}$$

where $E_p\Big[rawScore\Big]$ and $SD_p\Big[rawScore\Big]$ are the expectation and standard deviation in frequency bin p, respectively, of raw scores.

It is recommended to normalize all data together, therefore `selscan norm` accepts multiple `selscan` output files. Normalization of iHS or nSL output using 100 frequency bins can be performed with the following command.

```
selscan norm [--ihs|--nsl] --files *.out --bins 100
```

XPEHH and XPnSL scores (phased or unphased) and iHH12 scores are not correlated with derived allele frequency (see Section 3.1.1) and are therefore normalized by

$$score = \frac{rawScore - E\Big[rawScore\Big]}{SD\Big[rawScore\Big]}. \tag{14}$$

where $E\Big[rawScore\Big]$ and $SD\Big[rawScore\Big]$ are the expectation and standard deviation, respectively, of raw scores across the genome.

Normalization of XPEHH, XPnSL, or iHH12 output can be performed with the following command.

```
selscan norm [--ihh12|--xpehh|--xpnsl] --files *.out
```

Speeding Up Normalization. Sometimes it is desirable to normalize a set of data using a set of reference data. This is most commonly encountered when using neutral simulations as a background, either for normalizing empirical data or for normalizing individual simulation replicates. Repeatedly normalizing against the same reference data can be time-consuming. `selscan norm` accommodates this by providing a logging the normalization information, which can then be passed as parameter to be used for normalizing other output.

```
selscan norm [--ihs|--nsl|--ihh12|--xpehh|--xpnsl] --log neutral.log \
--files neutral_rep*.out
selscan norm [--ihs|--nsl|--ihh12|--xpehh|--xpnsl] --log-input neutral.log \
--files sweep_rep*.out
```

2.5.2 Window-based outlier detection

While individual extreme scores are suggestive of a sweep, it has been shown that searching for clusters of extreme scores increases power (Voight et al., 2006; Szpiech et al., 2021). To facilitate this type of analysis, `selscan norm` can perform window-based outlier detection, by searching for non-overlapping windows in the genome that contain a high percentage of "extreme" scores. The following command normalizes raw `selscan` output and performs an outlier analysis in non-overlapping 100kb windows.

```
selscan norm [--ihs|--nsl|--ihh12|--xpehh|--xpnsl] --files *.out --bp-win \
--winsize 100000
```

For iHS, nSL, and iHH12, this will compute the percentage of scores within each window for which $|score| > C$, for positive C. For XPEHH and XPnSL, this will make two calculations: the percentage of scores within each window for which $score > C$ and the percentage of scores for which $score < -C$. This allows windows to be identified which indicate an enrichment in either population. By default $C = 2$ as suggested in previous studies (Voight et al., 2006; Szpiech et al., 2021), however this value can be changed with `--crit-val`. Windows are binned into quantiles by

number of scores (default 10 bins, use `--qbins` to change) then annotated as being in the top 1% or top 5% within each quantile.

The top 1% windows are typically chosen as outliers when identifying candidate regions under selection. While strong sweep candidates often appear in the top 1%, signals at loci that are weaker may only become visible when using a broader threshold, such as the top 5%. `selscan norm` also provides an option to examine finer-grained percentiles by setting `--fine-percentile`. This allows users to annotate each window with an integer percentile from 1 to 100 (i.e., 1%, 2%, 3%, . . . , 100%).

2.5.3 Gene-based analyses

It is often desirable to examine which genes (or other genomic features) intersect with the outlier windows identified with `--bp-win` and to compute the maximum observed score per gene. The latter information may be used to rank genes or to perform enrichment tests. `selscan norm` can intersect and annotate windows and compute maximum observed scores per gene when given a BED file. If `features.bed` contains a list of intervals defining gene regions and IDs and `selscan.out.norm.100kb.windows` is the output from `selscan norm --bp-win`, the following command can be executed.

```
selscan norm [--ihs|--nsl|--ihh12|--xpehh|--xpnsl] \
--annotate-win selscan.out.norm.100kb.windows --gene-bed features.bed
```

This command outputs two files: (i) `selscan.out.norm.100kb.windows.ann`, which contains the windows annotated with overlapping gene names, and
(ii) `selscan.out.norm.100kb.windows.genetable`, which summarizes results at the gene level. The gene table has five columns: *gene name*, *gene length*, *number of overlapped windows*, *maximum score*, and *maximum score corrected for gene length*.

The final column is given by the residuals after regressing the score onto gene length. This is an important adjustment as maximum observed score is correlated with gene length (see Section 3.2.1), and gene length is correlated with many biologically interesting features (Lopes et al., 2021). To correct for potential dependence of scores on gene length, we perform a linear regression of the per-gene score on the logarithm of gene length:

$$\text{score}_i = \beta_0 + \beta_1 \cdot \log(\text{length}_i) + \epsilon_i,$$

where score_i is the score assigned to gene i, length_i is the length of the gene, β_0 and β_1 are the regression coefficients, and ϵ_i is the residual. The residuals ϵ_i are returned as length-corrected scores, allowing fair comparison of scores across genes of different lengths.

2.6 Additional options in selscan

`selscan` has many addition options for user convenience. For example, if a genetic map is unavailable but required for a particular statistic, the flag `--pmap` can be set so `selscan` uses physical distances instead. Legacy formats for genetic data are supported, such as TPED (Purcell et al., 2007), HAP (Baumdicker et al., 2022), and transposed HAP (Howie et al., 2009). Additionally, multiple runs of `selscan` can be executed with different parameter configurations and multiple statistics using the `--multi-param` option. For a full listing and explanation of additional user options, see the `selscan` manual at `https://github.com/szpiech/selscan`.

2.7 Simulated and Empirical Datasets

2.7.1 Simulated Data

Two simple demographic history models, a single population model and a two population divergence model, were simulated to illustrate how the EHH-based statistics implemented in `selscan v3.0` function. For each model, a single hard sweep was simulated, and this was performed both in the presence of background selection and on a purely neutral background. This created four scenarios in total: a hard sweep in a single population, a hard sweep in a single population with background selection, two populations with a hard sweep in one, and two populations with background selection with a hard sweep in one. All simulations were generated using `SLiM` (Haller et al., 2025) with a chromosome length of 10 Mb, a recombination rate of 1×10^{-8}, and a mutation rate 1.29×10^{-8}.

In the single population model, the effective population size was set to $N_e = 10{,}000$, and, after a $100,000$ generation burn-in, a beneficial mutation ($s = 0.1$) was introduced at position 5 Mb in the $100,001$st generation. The simulation ran for an additional $2,000$ generations or until the beneficial mutation reached $\sim 75\%$ frequency. If the beneficial mutation was lost the simulation

was restarted. At the end of a successful simulation, a VCF file was output. For the scenario with background selection deleterious mutations was introduced at a rate of $9:1$, neutral:deleterious, with selection coefficients drawn from a gamma distribution with mean 0.1 and shape 0.2. Data for normalization was generated in the same way but without introducing a beneficial allele. 100 replicates of normalization data were generated for each scenario (with and without background selection). Fifty individuals were sampled for each simulation. The resulting variants were filtered for biallelic SNPs and removed if the minor allele frequency below 0.05.

In the two population model, an ancestral population with effective population size $N_e = 10{,}000$ was simulated for a $100,000$ generation burn-in. At generation $100,500$, the ancestral population split into two populations each with effective population size $N_e = 10{,}000$. At generation $102,000$, a beneficial mutation ($s = 0.1$) was introduced into one population at position 5 Mb. The simulation ran for an additional $2,000$ generations or until the beneficial mutation reached $\sim 75\%$ frequency. If the beneficial mutation was lost the simulation was restarted. At the end of a successful simulation, a VCF file was output. For the scenario with background selection deleterious mutations was introduced at a rate of $9:1$, neutral:deleterious, with selection coefficients drawn from a gamma distribution with mean 0.1 and shape 0.2. Data for normalization was generated in the same way but without introducing a beneficial allele. 100 replicates of normalization data were generated for each scenario (with and without background selection). Fifty individuals were sampled for each simulation. The resulting variants were filtered for biallelic SNPs.

2.7.2 Empirical Data

The phased and imputed genomes of two populations from the 1000 Genomes Project were used: 99 Utah individuals with Northern and Western European ancestry (CEU) and 108 Yoruba individuals from Ibadan, Nigeria (YRI) (Consortium et al., 2015; Byrska-Bishop et al., 2022). All autosomes were extracted and were restricted to biallelic SNPs. For iHS, nSL, and iHH12 calculations, `selscan` filters sites with a minor allele frequency less than 0.05.

Gene annotations were obtained from the GENCODE database (GRCh38) using the GTF file from (Frankish et al., 2021).

3 Results

In this section, we summarize the outcomes of applying `selscan`, and subsequently, `selscan norm` to both simulated and empirical datasets.

3.1 Simulated Data

3.1.1 Importance of Normalization

Figures 1 and 2 illustrate the importance of normalization for iHS and nSL. In Figures 1a and 2a, the raw statistic from all simulated data is plotted against derived allele frequency, showing a clear correlation. This is because iHS and nSL are log-ratios of EHH_c (Equation 3), which is summarizing haplotype lengths surrounding either the ancestral or derived alleles. As low-frequency alleles are likely to be younger and therefore segregate on longer haplotypes, even under neutrality, this implies larger log-ratio scores for low frequency alleles. Figures 1b and 2b show how normalization in frequency bins (Equation 13) corrects for this frequency bias. To underscore the importance of normalization, Figures 1c and 2c contrast raw $|scores|$ and normalized $|scores|$ as a function of genome location for the sweep simulation, clearly demonstrating that without normalization the sweep signal would be lost.

XPEHH and XPnSL, however, do not suffer from this dependence on derived allele frequency, as they are comprised of the EHH (Equation 1) of all haplotypes within each population, instead of restricting to haplotypes carrying a particular allele. This is illustrated in Figure 3. We show similar behavior for iHH12 in Figure 4. However, normalization is still recommended, as it puts different experiments on the same scale and allows for easy downstream statistical analyses which may assume variables are distributed Normally.

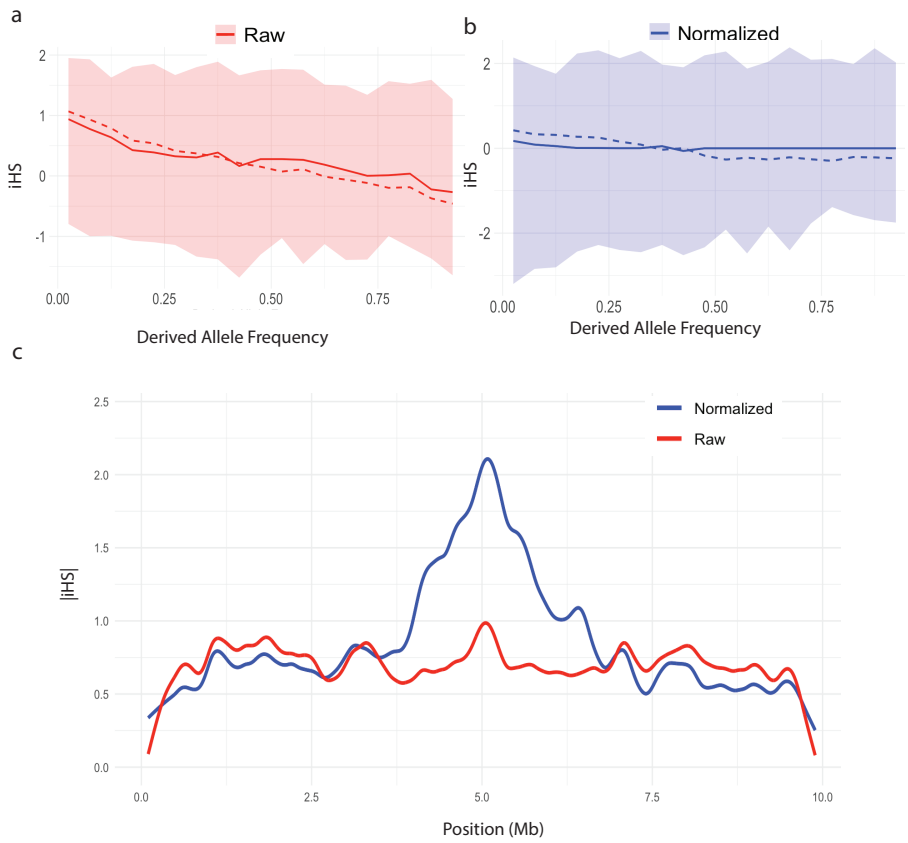


Figure 1: The mean (solid line), median (dashed line), and 1st–99th percentile rang (shaded region) for **(a)** raw iHS and **(b)** normalized iHS statistic plotted against derived allele frequency. **(c)** Spline smoothed raw and normalized curves of $|iHS|$ in the vicinity of a simulated hard sweep at 5Mb.

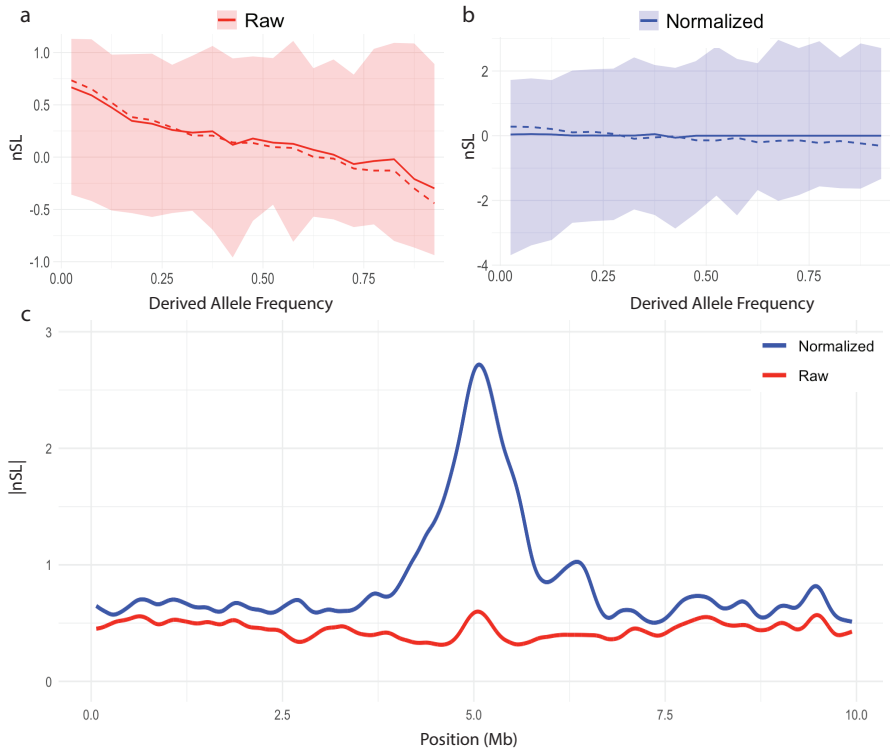


Figure 2: The mean (solid line), median (dashed line), and 1st–99th percentile rang (shaded region) for **(a)** raw nSL and **(b)** normalized nSL statistic plotted against derived allele frequency. **(c)** Spline smoothed raw and normalized curves of $|nSL|$ in the vicinity of a simulated hard sweep at 5Mb.

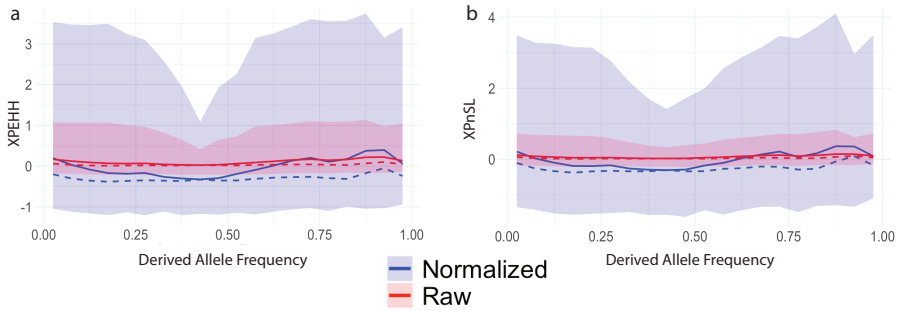


Figure 3: The mean (solid line), median (dashed line), and 1st–99th percentile rang (shaded region) for **(a)** raw (red) and normalized (blue) XPEHH and **(b)** raw (red) and normalized (blue) XPnSL statistics plotted against derived allele frequency.

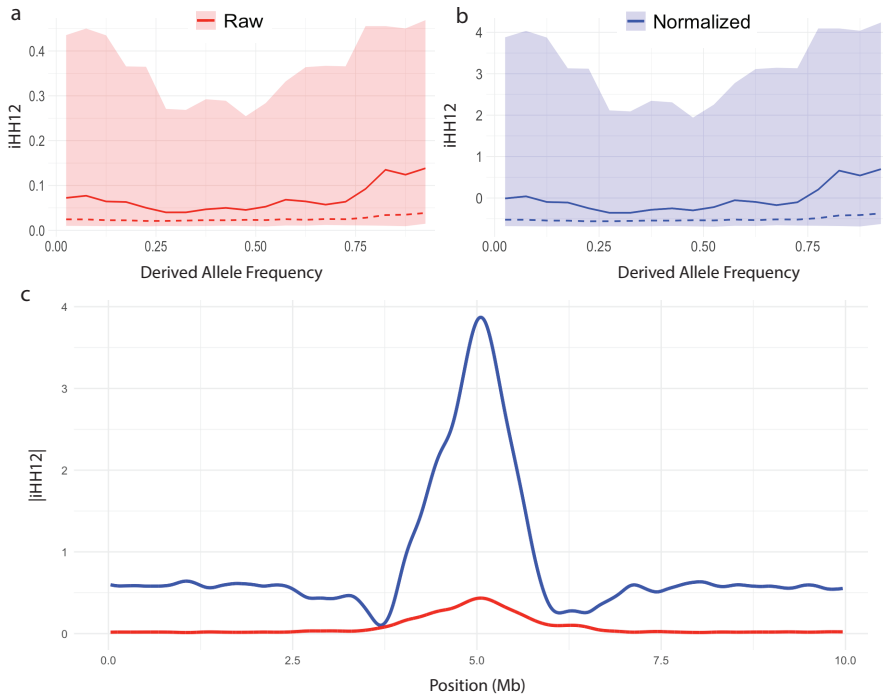


Figure 4: The mean (solid line), median (dashed line), and 1st–99th percentile rang (shaded region) for **(a)** raw iHH12 and **(b)** normalized iHH12 statistic plotted against derived allele frequency. **(c)** Spline smoothed raw and normalized curves of $|iHH12|$ in the vicinity of a simulated hard sweep at 5Mb.

3.1.2 Single-population Statistics

Figure 5 illustrates how EHH can distinguish between genome regions with an adaptive allele sweeping and genome regions without an adaptive allele. Figure 5a plots the EHH of the entire sample in the vicinity of a sweep (dotted line), which decays slowly over distance, in contrast to the EHH in a neutral region (solid line) which decays quickly. Figure 5b illustrates how computing EHH_c can distinguish between the sweeping haplotype and the non sweeping haplotype. In the sweep simulation, EHH_1 is computed among the haplotypes carrying the adaptive derived allele, and haplotype homozygosity decays slowly. However, in the same simulation at the same locus, EHH_0 is computed among the haplotypes not carrying the adaptive allele, and haplotype homozygosity decays quickly. Whereas in the neutral simulation both EHH_1 and EHH_0 decay quickly as neither set of hapotypes contain an adaptive allele. Figures 5c and Figure 5d show similar patterns in the context of background selection. Based on these patterns, iHH12, iHS, and nSL are designed to efficiently summarize the contrasts in these curves for large-scale querying of millions of loci across

a genome.

Figure 6 illustrates how iHH12 (Figure 6a), iHS (Figure 6b), and nSL (Figure 6c) can identify a sweep in the center of a simulated 10Mb region. For all three statistics, regardless of the presence of background selection and regardless of whether allele phase is utilized (iHS and nSL only), a clear peak forms in the vicinity of where the adaptive allele was placed. This suggests that the EHH-based single-population statistics are robust to background selection.

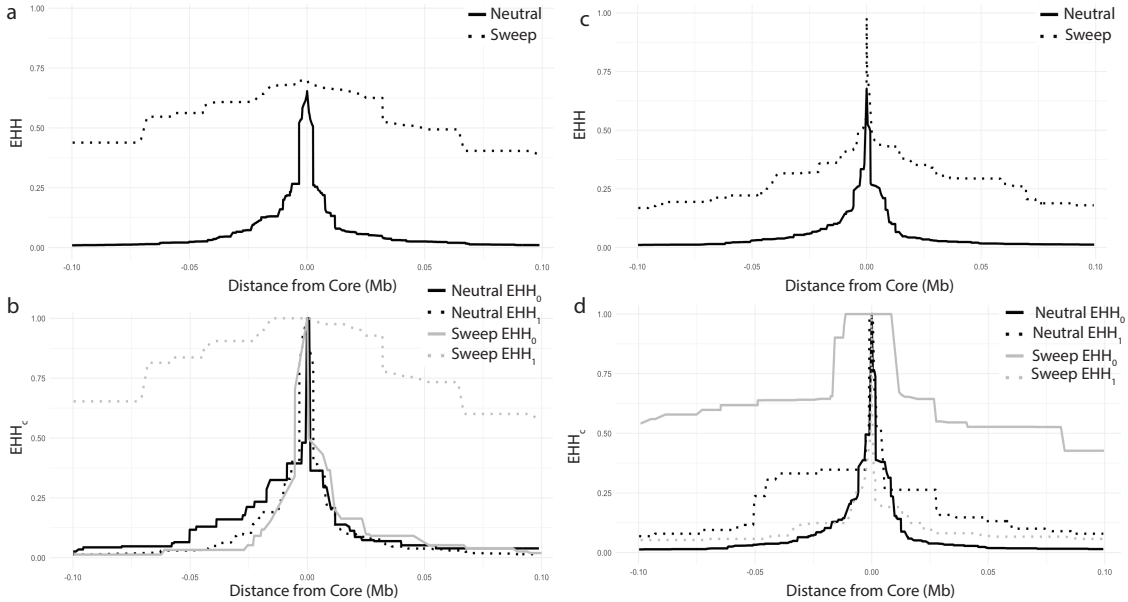


Figure 5: **(a)** EHH curves for a simulated hard sweep and a simulated neutral region. **(b)** EHH_c curves for each of the core haplotypes (0 and 1) for a simulated hard sweep and a simulated neutral region. **(c)** EHH curves for a simulated hard sweep and a simulated region without a sweep in the presence of background selection. **(d)** EHH_c curves for each of the core haplotypes (0 and 1) for a simulated hard sweep and a simulated region without a sweep in the presence of background selection.

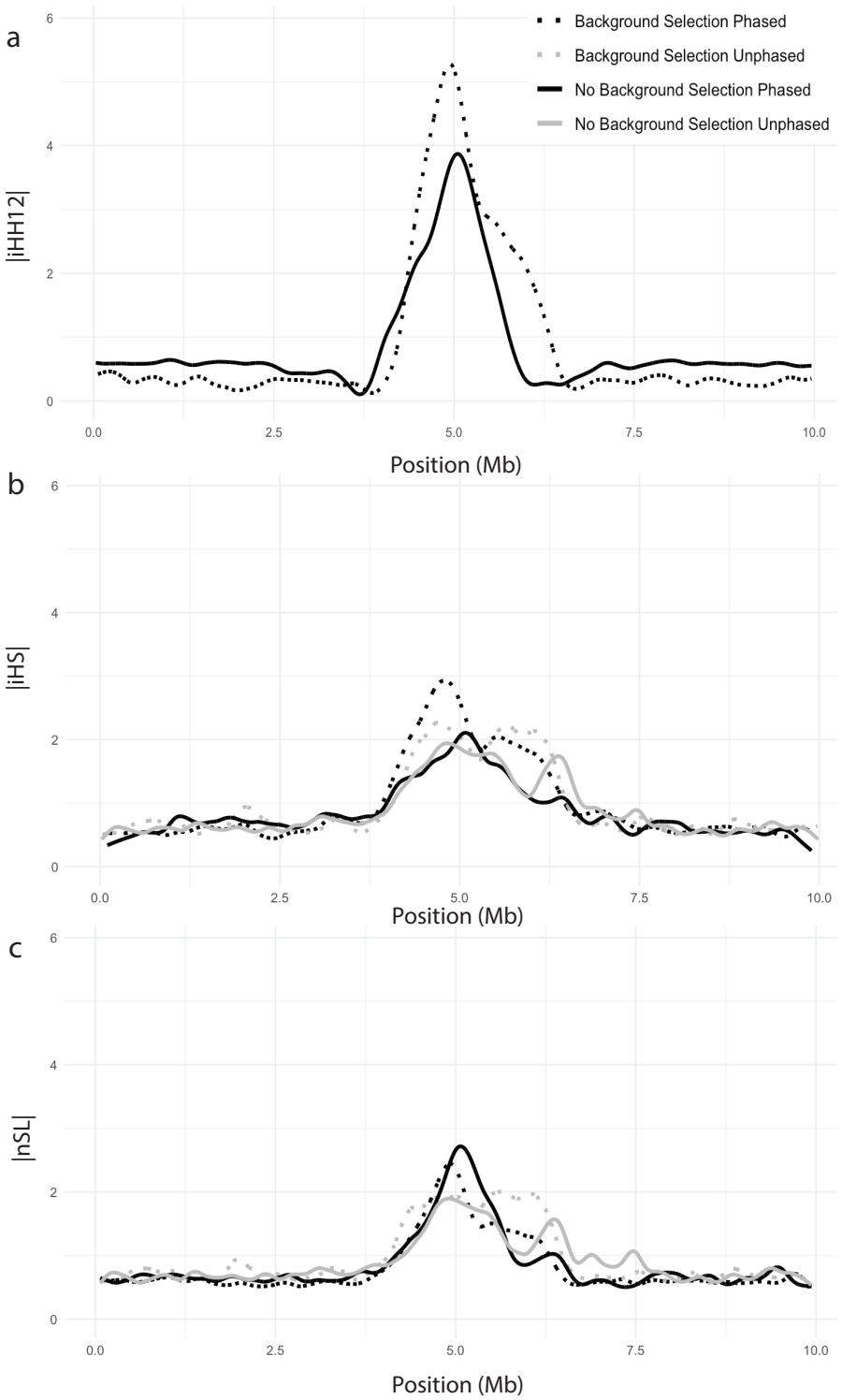


Figure 6: Normalized **(a)** $|iHH12|$, **(b)** $|iHS|$, and **(c)** $|nSL|$ plotted against genomic position for a simulated hard sweep with and without background selection. All curves were splined smoothed.

3.1.3 Two-population Statistics

The two-population statistics XPEHH and XPnSL are designed to summarize and contrast EHH patterns between two closely related populations to identify local adaptation. Figure 7 illustrates how these two statistics peak in the center of the simulated region where the adaptive allele was place. Figures 7a plot XPnSL and Figures 7b plot XPEHH. In addition to testing the influence of phased genotypes and background selection, the influence of unequal sample sizes among the two populations was considered, showing minimal influence. This suggests that the EHH-based two-population statistics are robust to background selection and uneven samples size.

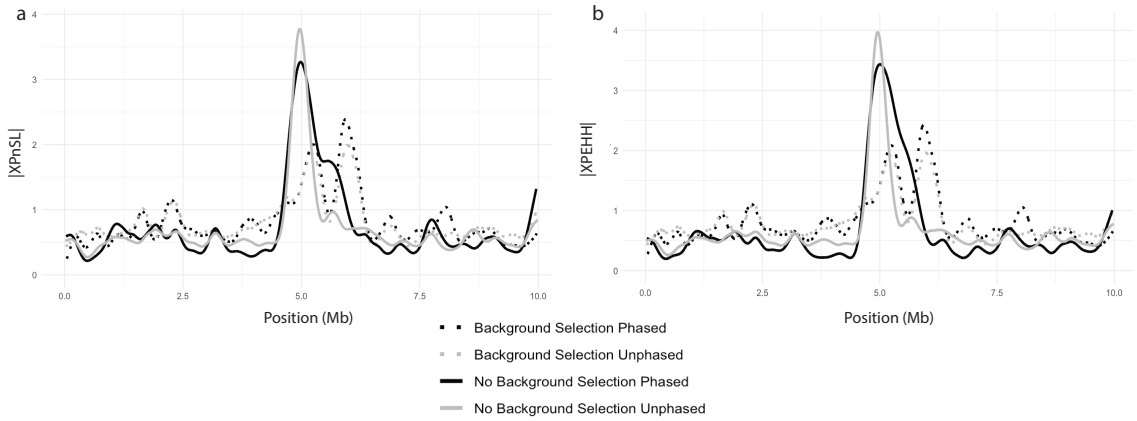


Figure 7: Normalized **(a)** XPnSL and **b** XPEHH plotted against genomic position for a simulated hard sweep with and without background selection. All curves were splined smoothed.

3.2 Basic downstream analysis with `selscan`

Section 3.1 demonstrates how `selscan`'s EHH-based statistics behave under simulated scenarios. However, `selscan`'s `norm` subcommand contains basic downstream analysis tools that are best illustrated using an empirical dataset.

This section illustrates an example of `selscan norm`'s window-based outlier detection and gene annotation functions can be used to find genes putatively under positive selection. This example uses the European CEU and Yoruba YRI populations from the 1000 Genomes Project (Consortium et al., 2015; Byrska-Bishop et al., 2022) and focuses on the well-known selection signal in the LCT/MCM6 region of the human genome.

It involves lactase persistence, a phenotype which has evolved multiple times in humans (Tishkoff

et al., 2007). In the CEU population, this phenotype is caused by a variant within an intron in the MCM6 gene upstream of LCT, allowing lactase expression into adulthood (Ingram et al., 2009).

`selscan` was used to compute iHS, nSL, iHH12 across all autosomes in the CEU population, and XPEHH and XPnSL were computed on all autosomes comparing CEU and YRI populations. Genome-wide normalization of all statistics was performed using `selscan norm`, with iHS and nSL being normalized with 100 frequency bins. Figure 8 shows the raw-score distributions (Figure 8a and c) and normalized-score distributions (Figure 8b and d) for scores on chromosome 2.

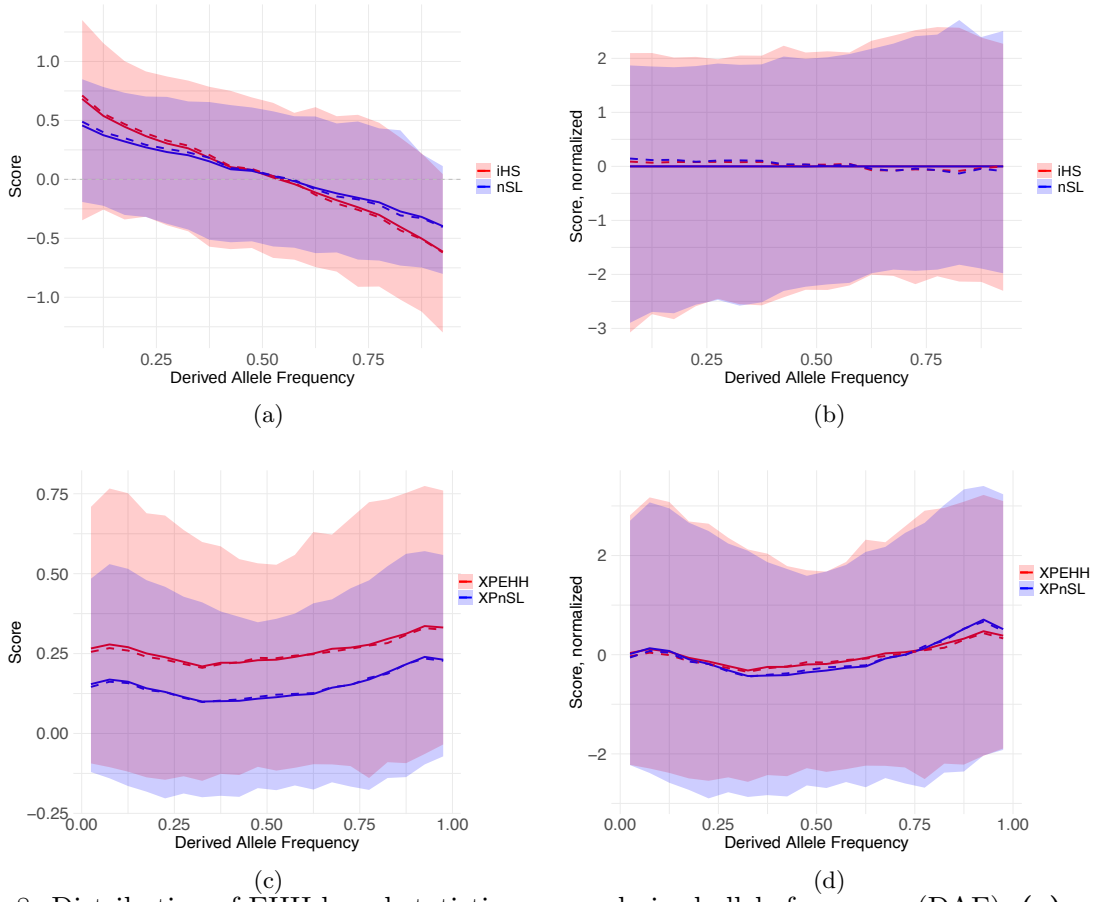


Figure 8: Distribution of EHH-based statistics versus derived allele frequency (DAF). **(a)** raw iHS (red) and nSL (blue) scores. **(b)** normalized iHS (red) and nSL (blue) scores. **(c)** raw XPEHH (red) and XPnSL (blue) scores. **(d)** normalized XPEHH (red) and XPnSL (blue) scores. All panels show the mean (solid line), median (dashed line), and 1st–99th percentile range (shaded area).

3.2.1 Analysis of chromosome 2 identifies the LCT/MCM6 locus

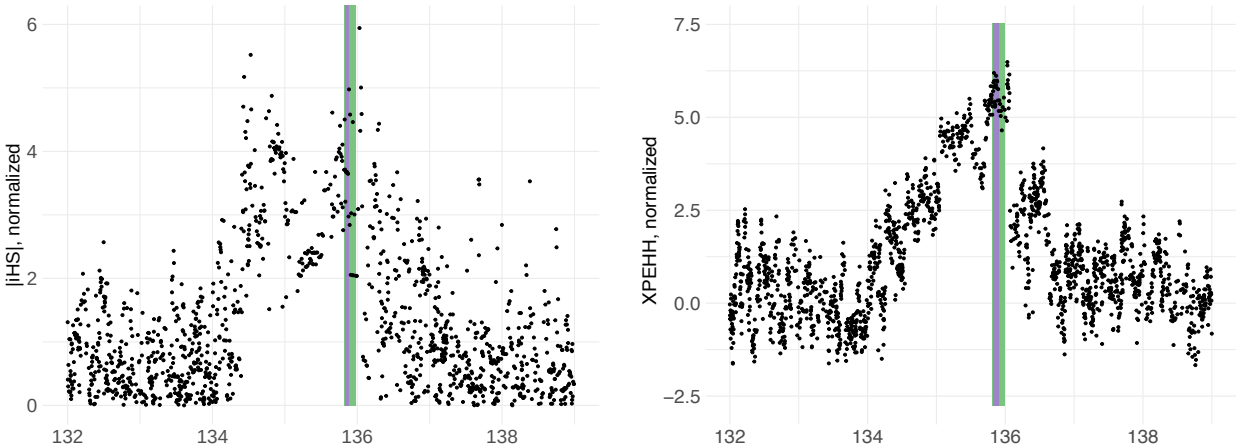
Focusing on chromosome 2 XPEHH scores, `selscan norm` was used to identify window-based outliers (default parameters) and annotated with overlapping gene regions. Table 1 shows `selscan`

`norm` output restricted to windows falling in the top 1% and sorted by maximum observed score per window. As anticipated, the LCT/MCM6 locus is found at the very top of this list, confirming that this locus is among the strongest selected on chromosome 2.

Table 1: Top 20 windows on chromosome 2 from the top 1% of XP-EHH scores (sorted by maximum score column) showing the maximum score per window and the genes overlapping each window.

Start (bp)	End (bp)	nSnps	Frac of SNPs with score above threshold	Maximum Score	Overlapped Genes
135,800,001	135,900,000	1,946	1.00	6.19	**LCT, MCM6**
135,700,001	135,800,000	1,838	1.00	5.86	R3HDM1, UBXN4, **LCT**
135,900,001	136,000,000	1,800	1.00	5.70	DARS1
135,400,001	135,500,000	2,288	1.00	5.51	ZRANB3
135,200,001	135,300,000	1,958	1.00	5.16	ZRANB3
135,300,001	135,400,000	2,016	1.00	5.10	ZRANB3
135,000,001	135,100,000	1,900	1.00	4.99	MAP3K19, RAB3GAP1
135,100,001	135,200,000	1,752	1.00	4.88	RAB3GAP1, ZRANB3
135,500,001	135,600,000	1,800	1.00	4.81	ZRANB3, R3HDM1
236,200,001	236,300,000	2,095	0.51	4.59	ASB18
237,500,001	237,600,000	2,822	0.56	4.29	MLPH, PRLH, RAB17
136,500,001	136,600,000	2,245	0.93	4.26	-
134,900,001	135,000,000	2,015	0.83	4.15	ACMSD, CCNT2, MAP3K19
237,400,001	237,500,000	2,309	0.72	4.07	COL6A3, MLPH
134,800,001	134,900,000	1,991	1.00	4.04	ACMSD
6,100,001	6,200,000	2,684	0.59	4.00	-
177,600,001	177,700,000	2,215	0.52	3.93	IFT70A, PDE11A
136,400,001	136,500,000	2,186	0.89	3.78	-
126,800,001	126,900,000	2,447	0.40	3.63	TEX51
236,500,001	236,600,000	2,361	0.90	3.61	IQCA1, ACKR3

Zooming into the LCT/MCM6 locus, Figure 9a plots normalized $|iHS|$ as a function of genome position on chromosome 2, and Figure 9b plots normalized $XPEHH$ scores, as a function of genome position on chromosome 2. Both plots show broad peaks overlapping the LCT/MCM6, suggesting a clear signal of positive selection at this locus.



(a) (b)

Figure 9: Normalized **(a)** $|iHS|$ plotted against chromosome 2 position in CEU, and normalized **(b)** XPEHH between CEU and YRI plotted against chromosome 2 position. The green shaded area highlights LCT, and the purple shaded area highlights MCM6.

`selscan norm` also creates a gene table (Table 2) for each gene. This table provides information on a per-gene basis including gene length (as given in the BED file), number of overlapping windows, max observed score in the gene, and an adjusted max observed score. The latter is provided to account for varying gene length. Figure 10a shows the correlation of XPEHH score with gene length, and Figure 10b shows the relationship after regressing out its influence. Table 2 is sorted by length-corrected maximum observed score, and LCT and MCM6 fall at the very top.

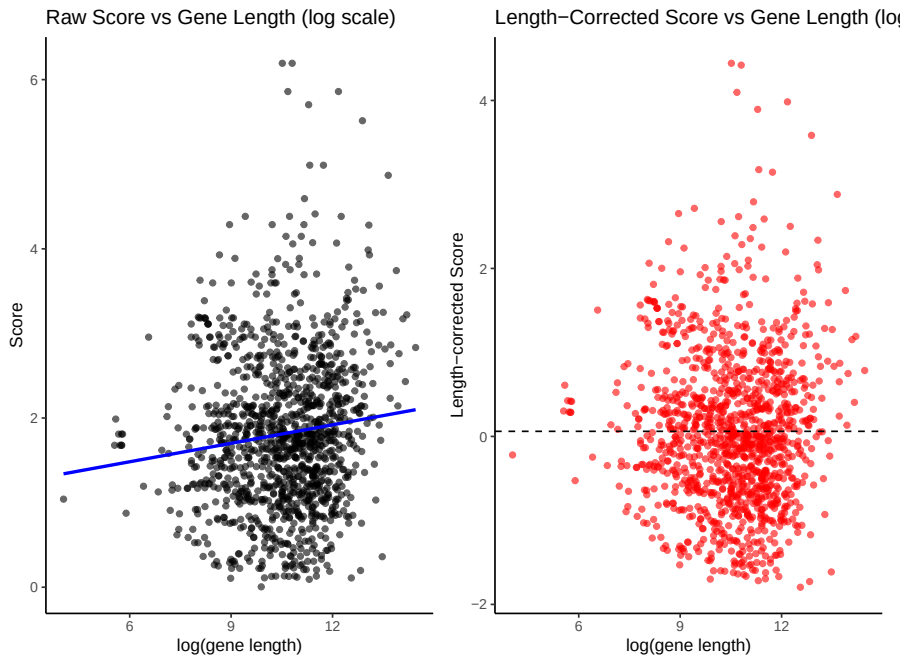


Figure 10: Length correction for XP-EHH scores in chromosome 2 (only positive values are shown). Left panel: raw per-gene scores plotted against log(gene length), with a linear regression trend line (blue) showing the overall relationship. Right panel: length-corrected scores (residuals from the regression) plotted against logarithm of gene length, with a dashed line indicating the mean.

Table 2: Normalized XP-EHH (CEU vs YRI) was computed in non-overlapping 100 kb genomic windows. For each window we averaged XP-EHH scores for SNPs falling inside the window and assigned the window to any gene that overlapped it (a gene can receive multiple windows). Using these 100 kb windows as the unit of analysis, LCT and the gene MCM6, which contains an enhancer region about 14kb upstream of LCT that regulates lactase expression, emerge as the top candidates on chromosome 2. Top genes overlapping high-scoring windows with length-corrected scores.We filtered windows to include only those with at least 10 SNPs, selected the top 1% based on score, and report the top 20 highest-scoring genes.

Gene Name	Length (bp)	# Overlapped Windows	Max Score	Length-Corrected Max Score
MCM6	36,854	1	6.2	4.4
LCT	49,335	2	6.2	4.4
UBXN4	43,433	1	5.9	4.1
R3HDM1	193,867	3	5.9	4.0
DARS1	80,220	1	5.7	3.9
ZRANB3	394,313	5	5.5	3.6
MAP3K19	82,984	2	5.0	3.2
RAB3GAP1	124,385	2	5.0	3.1
LRRTM4	845,635	9	4.9	2.9
ASB18	70,951	2	4.6	2.8
SMIM39	12,326	1	4.4	2.7
PRLH	7,734	1	4.3	2.7
FAM168B	45,612	1	4.4	2.6
ANKRD36B	97,215	2	4.4	2.6
RAB17	27,302	2	4.3	2.6
ARHGEF4	210,350	3	4.4	2.5
MLPH	69,895	2	4.3	2.5
CCNT2	41,108	1	4.1	2.4
ACMSD	64,595	2	4.1	2.4
MYO3B	477,027	6	4.3	2.3

4 Discussion

The use of selection statistics is ubiquitous in modern evolutionary genomics. The ability to identify putatively selected loci is of great importance in forming our understanding of adaptation of organisms to myriad selection pressures (Fu and Akey, 2013). Therefore, the computation and analysis of selection statistics should be consistent across studies to maintain reproducibility of results and to ensure robust and reliable inference.

Many of the most widely used selection statistics for recombining genomes are based on Expected Haplotype Homozygosity (EHH), which measures the decay of haplotype identity away from a locus

of interest (Sabeti et al., 2002, 2007). Here we provide the formal definition of EHH and the formal definitions of the EHH-based statistics iHS, nSL, iHH12, XPEHH, and XPnSL. All statistics and their unphased equivalents are implemented in the software `selscan`. We also introduced the `selscan v3.0` subcommand `norm`. `selscan norm` is used it to perform essential normalization computations, and can also be used for window-based outlier detection and gene-based annotation.

We demonstrated the use of `selscan` on both simulated and empirical genetic data. Under ideal parameters, simulations illustrated the statistics' ability to identify loci under positive selection in the presence and absence of background selection. This supports the use of statistics based on EHH, since other selection statistics based on the site frequency spectrum can fail to uncover signals of positive selection in this context (but see (Huber et al., 2016; DeGiorgio et al., 2016). Further, we used `selscan norm` to show how, without normalization, the selection signal is dampened. Finally, we performed a downstream analysis with `selscan norm` on data from the 1000 Genomes Project, focusing on the well-known selection signal at the LCT/MCM6 locus in European individuals (Tishkoff et al., 2007). We showed how the results produced by `selscan` and `selscan norm` quickly produce window-level and gene-level data that can be presented as-is or used in further statistical analyses.

Ultimately, we hope to create a resource for the computation and interpretation of EHH-based selection statistics that is accessible to readers from various backgrounds. While `selscan` is designed to be user friendly, we note that the efficiency of `selscan`'s core algorithm in conjunction with `selscan norm` offers the rapid computation and analysis of statistics at millions of loci (Rahman et al., 2025). This offers a potential future avenue to scale EHH-based selection statistics to massive empirical and synthetic datasets, and for use in the training of machine learning models (Amin et al., 2024).

Acknowledgments

Computations for this research were performed using the Pennsylvania State University's Institute for Computational Data Sciences' Roar supercomputer. This work was supported by the National Institute of General Medical Sciences of the National Institutes of Health award number R35GM146926, and start-up funds from the Pennsylvania State University's Department of

Biology.

References

Amin, M. R., Hasan, M., and DeGiorgio, M. (2024). Digital image processing to detect adaptive evolution. *Molecular Biology and Evolution*, 41(12):msae242.

Arnab, S. P., Campelo dos Santos, A. L., Fumagalli, M., and DeGiorgio, M. (2025). Efficient detection and characterization of targets of natural selection using transfer learning. *Molecular Biology and Evolution*, 42(5):msaf094.

Baumdicker, F., Bisschop, G., Goldstein, D., Gower, G., Ragsdale, A. P., Tsambos, G., Zhu, S., Eldon, B., Ellerman, E. C., Galloway, J. G., et al. (2022). Efficient ancestry and mutation simulation with msprime 1.0. *Genetics*, 220(3):iyab229.

Benton, M. L., Abraham, A., LaBella, A. L., Abbot, P., Rokas, A., and Capra, J. A. (2021). The influence of evolutionary history on human health and disease. *Nature Reviews Genetics*, 22(5):269–283.

Booker, T. R., Jackson, B. C., and Keightley, P. D. (2017). Detecting positive selection in the genome. *BMC biology*, 15(1):98.

Byrska-Bishop, M., Evani, U. S., Zhao, X., Basile, A. O., Abel, H. J., Regier, A. A., Corvelo, A., Clarke, W. E., Musunuri, R., Nagulapalli, K., et al. (2022). High-coverage whole-genome sequencing of the expanded 1000 genomes project cohort including 602 trios. *Cell*, 185(18):3426–3440.

Consortium, . G. P. et al. (2015). A global reference for human genetic variation. *Nature*, 526(7571):68.

DeGiorgio, M., Huber, C. D., Hubisz, M. J., Hellmann, I., and Nielsen, R. (2016). Sweepfinder2: increased sensitivity, robustness and flexibility. *Bioinformatics*, 32(12):1895–1897.

DeGiorgio, M. and Szpiech, Z. A. (2022). A spatially aware likelihood test to detect sweeps from haplotype distributions. *PLoS genetics*, 18(4):e1010134.

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(5):1275–1291.

Frankish, A., Diekhans, M., Jungreis, I., Lagarde, J., Loveland, J. E., Mudge, J. M., Sisu, C., Wright, J. C., Armstrong, J., Barnes, I., et al. (2021). Gencode 2021. *Nucleic acids research*, 49(D1):D916–D923.

Fu, W. and Akey, J. M. (2013). Selection and adaptation in the human genome. *Annual review of genomics and human genetics*, 14(1):467–489.

Garud, N. R., Messer, P. W., Buzbas, E. O., and Petrov, D. A. (2015). Recent selective sweeps in north american drosophila melanogaster show signatures of soft sweeps. *PLOS Genetics*, 11(2):1–32.

Gautier, M., Klassmann, A., and Vitalis, R. (2017). rehh 2.0: a reimplementation of the r package rehh to detect positive selection from haplotype structure. *Molecular ecology resources*, 17(1):78–90.

Gautier, M. and Vitalis, R. (2012). rehh: an r package to detect footprints of selection in genome-wide snp data from haplotype structure. *Bioinformatics*, 28(8):1176–1177.

Haller, B. C., Ralph, P. L., and Messer, P. W. (2025). Slim 5: Eco-evolutionary simulations across multiple chromosomes and full genomes. *bioRxiv*, pages 2025–08.

Harris, A. M. and DeGiorgio, M. (2020). A likelihood approach for uncovering selective sweep signatures from haplotype data. *Molecular biology and evolution*, 37(10):3023–3046.

Hejase, H. A., Mo, Z., Campagna, L., and Siepel, A. (2022). A deep-learning approach for inference of selective sweeps from the ancestral recombination graph. *Molecular Biology and Evolution*, 39(1):msab332.

Howie, B. N., Donnelly, P., and Marchini, J. (2009). A flexible and accurate genotype imputation method for the next generation of genome-wide association studies. *PLoS genetics*, 5(6):e1000529.

Huber, C. D., DeGiorgio, M., Hellmann, I., and Nielsen, R. (2016). Detecting recent selective sweeps while controlling for mutation rate and background selection. *Molecular Ecology*, 25(1):142–156.

Ingram, C. J., Mulcare, C. A., Itan, Y., Thomas, M. G., and Swallow, D. M. (2009). Lactose digestion and the evolutionary genetics of lactase persistence. *Human genetics*, 124:579–591.

Kern, A. D. and Schrider, D. R. (2018). diplos/hic: an updated approach to classifying selective sweeps. *G3: Genes, Genomes, Genetics*, 8(6):1959–1970.

Klassmann, A. and Gautier, M. (2022). Detecting selection using extended haplotype homozygosity (ehh)-based statistics in unphased or unpolarized data. *PloS one*, 17(1):e0262024.

Lopes, I., Altab, G., Raina, P., and de Magalhães, J. P. (2021). Gene size matters: An analysis of gene length in the human genome. *Frontiers in Genetics*, Volume 12 - 2021.

Maclean, C. A., Chue Hong, N. P., and Prendergast, J. G. (2015). Hapbin: an efficient program for performing haplotype-based scans for positive selection in large genomic datasets. *Molecular biology and evolution*, 32(11):3027–3029.

Nielsen, R., Williamson, S., Kim, Y., Hubisz, M. J., Clark, A. G., and Bustamante, C. (2005). Genomic scans for selective sweeps using snp data. *Genome research*, 15(11):1566–1575.

Pavlidis, P. and Alachiotis, N. (2017). A survey of methods and tools to detect recent and strong positive selection. *Journal of Biological Research-Thessaloniki*, 24(1):7.

Purcell, S., Neale, B., Todd-Brown, K., Thomas, L., Ferreira, M. A., Bender, D., Maller, J., Sklar, P., De Bakker, P. I., Daly, M. J., et al. (2007). Plink: a tool set for whole-genome association and population-based linkage analyses. *The American journal of human genetics*, 81(3):559–575.

Rahman, A., Smith, T. Q., and Szpiech, Z. A. (2025). Fast and memory-efficient dynamic programming approach for large-scale ehh-based selection scans. *bioRxiv*, pages 2025–04.

Sabeti, P. C., Reich, D. E., Higgins, J. M., Levine, H. Z., Richter, D. J., Schaffner, S. F., Gabriel, S. B., Platko, J. V., Patterson, N. J., McDonald, G. J., et al. (2002). Detecting recent positive selection in the human genome from haplotype structure. *Nature*, 419(6909):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., et al. (2007). Genome-wide detection and characterization of positive selection in human populations. *Nature*, 449(7164):913–918.

Scheinfeldt, L. B. and Tishkoff, S. A. (2013). Recent human adaptation: genomic approaches, interpretation and insights. *Nature Reviews Genetics*, 14(10):692–702.

Stern, A. J., Wilton, P. R., and Nielsen, R. (2019). An approximate full-likelihood method for inferring selection and allele frequency trajectories from dna sequence data. *PLoS Genetics*, 15(9):e1008384.

Sugden, L. A., Atkinson, E. G., Fischer, A. P., Rong, S., Henn, B. M., and Ramachandran, S. (2018). Localization of adaptive variants in human genomes using averaged one-dependence estimation. *Nature communications*, 9(1):703.

Szpiech, Z. and Hernandez, R. (2016). Selective sweeps. In *Encyclopedia of Evolutionary Biology*, pages 23–32. Elsevier Inc.

Szpiech, Z. A. (2024). selscan 2.0: scanning for sweeps in unphased data. *Bioinformatics*, 40(1):btae006.

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(10):2824–2827.

Szpiech, Z. A., Novak, T. E., Bailey, N. P., and Stevison, L. S. (2021). Application of a novel haplotype-based scan for local adaptation to study high-altitude adaptation in rhesus macaques. *Evolution letters*, 5(4):408–421.

Tishkoff, S. A., Reed, F. A., Ranciaro, A., Voight, B. F., Babbitt, C. C., Silverman, J. S., Powell, K., Mortensen, H. M., Hirbo, J. B., Osman, M., et al. (2007). Convergent adaptation of human lactase persistence in africa and europe. *Nature genetics*, 39(1):31–40.

Torres, R., Szpiech, Z. A., and Hernandez, R. D. (2018). Human demographic history has amplified the effects of background selection across the genome. *PLOS Genetics*, 14(6):1–27.

Vaughn, A. H. and Nielsen, R. (2024). Fast and accurate estimation of selection coefficients and allele histories from ancient and modern dna. *Molecular biology and evolution*, 41(8):msae156.

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(3):e72.