SelNeTime: a python package inferring effective population size and selection intensity from genomic time series data

Mathieu Uhl[1,2,*], Paul Bunel[1,3,*], Miguel de Navascués[1], Simon Boitard[1,*], Bertrand Servin[3,*]

[1]CBGP, Université de Montpellier, CIRAD, INRAE, IRD, Institut Agro Montpellier, Montpellier, France
[2]CEFE, CNRS, Montpellier, France
[3]GenPhySE, Université de Toulouse, INRAE, INPT, Castanet-Tolosan, France
[*]co-first and co-last authors

October 30, 2025

Summary

Genomic samples collected from a single population over several generations provide direct access to the genetic diversity changes occurring within a specific time period. This provides information about both demographic and adaptive processes acting on the population during that period. A common approach to analyze such data is to model observed allele counts in finite samples using a Hidden Markov Model (HMM) where hidden states are true allele frequencies over time (i.e. a trajectory). The HMM framework allows one to compute the full likelihood of the data, while accounting both for the stochastic evolution of population allele frequencies along time and for the noise arising from sampling a limited number of individuals at possibly spread out generations. Several such HMM methods have been proposed so far, differing mainly in the way they model the transition probabilities of the Markov chain. Following Paris et al. (2019a), we consider here the Beta with Spikes approximation, which avoids the computational issues associated to the Wright-Fisher model while still including fixation probabilities, in contrast to other standard approximations of this model like the Gaussian or Beta distributions. To facilitate the analysis and exploitation of genomic time series data, we present an improved version of Paris et al. (2019a) 's approach, denoted SelNeTime, whose computation time is drastically reduced and which accurately estimates effective population size (assuming no selection) or the selection intensity at each locus (given a previously estimated value of N). We also evaluate the performance of this method in

realistic situations where selection is present and both demography and selection need to be inferred. SelNeTime is implemented in a user friendly python package, which can also easily simulate genomic time series data under a user-defined evolutionary model and sampling design.

Introduction

One limitation of most population genetic analyses is the need to use data from a single time point, the present. The genetic diversity observed in this single sample is very informative about past population history, but it integrates the effect of many evolutionary processes acting on a large time-scale, which may be hard to clearly separate. For instance, selection constraints acting on a given locus often fluctuate along time as a result of environmental changes, and such complex and changing processes are difficult to reconstruct from the observation of the resulting genetic diversity at a single recent time point. In contrast, the study of samples collected at several different generations allows one to directly measure changes in the genetic diversity of populations, informing about the evolutionary processes acting on that specific time period. Such genomic time series arise in different contexts and time scales, such as the monitoring of wild or domestic populations (Boitard et al., 2023), experimental evolution (Long et al., 2015) or ancient DNA (Fages Antoine et al., 2019) studies.

A common approach to analyze genomic time series data is to model observed allele frequencies using a Hidden Markov Model (HMM); this approach allows one to compute the full likelihood of the data, while accounting both for the stochastic evolution of population allele frequencies along time, under the combined effects of demography and selection, and for the noise arising from sampling a limited number of individuals at different generations (Bollback et al., 2008). Several such HMM methods have been proposed so far, differing mainly in the way they model the transition probabilities of the Markov chain. Indeed, while the Wright-Fisher (WF) model appears as a natural choice to model these transitions, it leads to transition matrices whose dimension is equal to the population size so it is limited in practice to small populations. Among the various approximations of the allele frequency transition distribution that have been proposed to overcome this issue (e.g. Gaussian, Beta), the Beta with Spikes (BwS) distribution (Tataru et al., 2015) was pointed out by Paris et al. (2019a) as particularly promising, because it combines a continuous (Beta) density with discrete fixation probabilities at frequencies 0 and 1. These features provide accurate estimations for a large range of demographic and selective parameters (Paris et al., 2019a), while previous methods were either impossible to run in large populations or inaccurate for some parts of the parameter space (e.g. strong selection).

However, the current implementation of this method, compareHMM (Paris et al., 2019b), considers a fixed effective population size provided by the user, assuming that this parameter is known or has been previously estimated from another software. Besides, it was initially implemented as a test tool comparing different approximations of the Wright-Fisher model in the context of genomic time series analyses, not to be used by a large community. In particular, it was not optimized to analyze very large data sets such as those typically found

in the genomics era.

With the objective of improving on these several limitations, the SelNeTime python package (Uhl et al., 2025) provides a user friendly and computationally efficient implementation of Paris et al. (2019a)'s method, which estimates effective population size or selection from genomic time series. SelNeTime can also simulate such data in order to easily test the expected performance of the approach under custom sampling designs. This package is available on the python package index https://pypi.org/project/selnetime/ and on the project git repository https://forge.inrae.fr/genetic-time-series/selnetime.

Materials and Methods

Modelling genomic time series through an HMM

Let us consider a population for which genomic data has been collected at n sampling times $t_1, \ldots, t_n$. For a given bi-allelic locus, we model the evolution of the (arbitrary) reference allele frequency along time using a Hidden Markov Model (HMM). We denote X_i the (hidden) population allele frequency at time t_i and Y_i the (observed) sample allele frequency at time t_i. As all HMMs, this stochastic process is fully determined by three components : the initial distribution of X_1, the transition distributions from X_{i-1} to X_i ($i \in 2, \ldots, n$) and the emission distributions of Y_i given X_i ($i \in 1, \ldots, n$). Here we assume that X_1 is distributed uniformly in $[0, 1]$ and that observed frequencies result from a binomial sampling within the population, i.e.

$$e_i(y_i; x_i) = P(Y_i = y_i \mid X_i = x_i) = C_{n_i}^{y_i}(x_i)^{y_i}(1 - x_i)^{n_i - y_i}$$

where y_i is the number of reference allele copies observed at time t_i and n_i the haploid sample size (i.e. total number of allele copies) at this time. Transition distributions, which make the link between observed data and the parameters of interest, are described in the next two sections.

Modelling transitions through the Wright-Fisher model

Let us now assume that the studied population evolves under a Wright-Fisher (WF) model, i.e. with panmixia, non overlapping generations and a constant (haploid) size N. Under this model, allele frequencies X_i are discrete and take values in $\{0, \frac{1}{N}, \ldots, 1\}$. We ignore mutations due to the short time scale of interest, but model selection using a diploid codominant fitness function with selection coeficient s, i.e. the fitnesses of homozygote reference, heterozygote and homozygote alternative genotypes are respectively equal to 1, $1 + s/2$ and $1 + s$. For a single generation, this results in the transition probabilities

$$P(X_{t+1} = \frac{k}{N} | X_t = \frac{j}{N}) = C_N^k f_s(\frac{j}{N})^k (1 - f_s(\frac{j}{N}))^{N-k}, \quad j, k \in 0, 1, \ldots, N$$

where

$$f_s(x) = \frac{x(1 + s/2 + sx/2)}{1 + sx}$$

Denoting $Q_{s,N} = ((Q_{s,N})_{j,k})_{0 \leq j,k \leq N}$ the matrix of these single generation transition probabilities, $Q_{s,N,i}$ the transition probabilities from t_{i-1} to t_i can be computed as

$$P(X_i = \frac{k}{N} | X_{i-1} = \frac{j}{N}) = (Q_{s,N,i})_{j,k} = (Q_{s,N}^{t_i - t_{i-1}})_{j,k}$$

The Beta with Spikes approximation

Because $Q_{s,N}$ has dimension $(N+1) \times (N+1)$, likelihood computations under the WF model described above are only possible for small values of N. Following Paris et al. (2019a), SelNeTime computes (by default) this transition matrix based on a Beta with Spikes (BwS) approximation of the WF model (Tataru et al., 2015). This model assumes that the probability density of X_i is given by

$$g_i(x) = p_{0,i}\delta(x) + p_{1,i}\delta(1-x) + (1 - p_{0,i} - p_{1,i})B(x; a_i, b_i)$$

where $p_{0,i}$ and $p_{1,i}$ are the fixation probabilities in 0 and 1 at time t_i, $\delta(x)$ the Dirac delta function and $B(x; a_i, b_i)$ is the probability density of a beta distribution with parameters a_i and b_i. While the continuous part modelled by $B()$ allows one to control the size of $Q_{s,N,i}$ using a discretization grid that is independent of N, accounting for the discrete fixation probabilities $p_{0,i}$ and $p_{1,i}$ was shown to provide better results compared to other standard continuous approximations of the WF (Paris et al., 2019a).

Following Tataru et al. (2015), the transition matrix $Q_{s,N,i}$ from time t_{i-1} to time t_i is computed in three steps. 1) We compute $m_i(N, s, x) = E[X_i \mid X_{i-1} = x]$ and $z_i^2(N, s, x) = Var(X_i \mid X_{i-1} = x)$ under a Wright-Fisher process, using a recursion from $t = 1$ to $t = t_i - t_{i-1}$. For $s \neq 0$, this recursion relies on a large N Taylor expansion called Delta method (Lacerda & Seoighe, 2014). 2) Using another recursion from $t = 1$ to $t = t_i - t_{i-1}$, we set the values $p_{0,i}(N, s, x)$, $p_{1,i}(N, s, x)$, $a_i(N, s, x)$ and $b_i(N, s, x)$ ensuring that the mean and variance of the resulting BwS distribution match the values $m_i(N, s, x)$ and $z_i^2(N, s, x)$ obtained at step 1. 3) The line of $Q_{s,N,i}$ corresponding to a starting frequency of x is derived from the BwS distribution characterized at step 2. Detailed equations used at steps (1-2) can be found in Paris et al. (2019a).

For step 3, we introduced a technical difference with the approach of Paris et al. (2019a). In the original study, the quantities stored in $Q_{s,N,i}$ were a mixture of probability and density values: the first and last columns represented the discrete probabilities of arriving respectively in 0 and 1, but other columns represented probability densities at arriving frequencies, which were later (numerically) integrated during likelihood computation. While more elegant mathematically, this approach occasionaly lead to numerical issues and implied that likelihood computations could not be fully vectorized, because different parts of the matrix were processed differently. In contrast, SelNeTime implementation considers that all coefficients in the matrix correspond to discrete probabilities. More precisely, for a discretization grid of $(0, 1)$ into n_x bins $I_j = [m_j, M_j]$, $j \in 1, \ldots, n_x$, $Q_{s,N,i}$, has dimension $(n_x + 2) \times (n_x + 2)$ where I_0 corresponds to an allele frequency of exactly 0 and I_{n_x+1} to an allele frequency of exactly 1. Its coefficients are defined by

$$(Q_{s,N,i})_{j,k} = P(X_i \in I_k | X_{i-1} \in I_j)$$

For $j \in 1, \ldots, n_x$ (i.e. when starting from a discretization bin), this probability is approximated by $P(X_i \in I_k | X_{i-1} = \overline{I_j})$, where $\overline{I_j}$ is the middle of I_j. For $k \in 1, \ldots, n_x$ (i.e. when arriving into a discretization bin), the probability is derived as $G(M_k) - G(m_k)$, where $G()$ is the cumulative density function of the Beta distribution obtained at step (2).

Likelihood computation and optimization

For an initial distribution ν, transition matrices $Q = (Q_2, \ldots, Q_n)$ and emission distributions $e = (e_1, \ldots, e_n)$, the likelihood of observed data in a HMM can be computed using the so-called forward algorithm. This algorithm focuses on the joint probability of observations Y_1 to Y_i and hidden state X_i

$$\alpha_i(x_i) = P(Y_1 = y_1, \ldots, Y_i = y_i, X_i = x_i)$$

Note that the $\alpha_i(x_i)$'s are discrete probabilities (rather than continuous measures) here, consistent with the above assumption of discretized allele frequencies. The algorithm is initiated by

$$\alpha_1(x_1) = e_1(y_1; x_1)\nu(x_1)$$

for all possible values of x_1 and then updated recursively, for $i = 2, \ldots, n$, by

$$\alpha_i(x_i) = e_i(y_i; x_i) \sum_{x_{i-1}} Q_i(x_{i-1}, x_i)\alpha_{i-1}(x_{i-1})$$

for all possible values of x_i. The notation $Q_i(x_{i-1}, x_i)$ is used here for convenience to refer to the entry of Q_i whose row corresponds to x_{i-1} and whose column corresponds to x_i. The likelihood is finally obtained by summing over x_n:

$$L(y_1, \ldots, y_n; \nu, Q, e) = P(Y_1 = y_1, \ldots, Y_n = y_n \mid \nu, Q, e) = \sum_{x_n} \alpha_n(x_n)$$

In our application, the parameters of interest are N and s, which influence Q but not ν and e. Thus, the forward algorithm provides

$$L(y_1, \ldots, y_n; \nu, Q, e) = L(y_1, \ldots, y_n; \nu, Q_{s,N,2}, \ldots, Q_{s,N,n}, e) = L(y_1, \ldots, y_n; N, s)$$

In other words, we are able to evaluate numerically the likelihood of the data for any given values of N and s. Note that this likelihood function also implicitely depends on the time intervals $t_i - t_{i-1}$, because these affect the $Q_{s,N,i}$ matrices (see section 'Modelling transitions through the Wright-Fisher model'). We omit this dependence here because we consider these parameters as known fixed values, but this could in principle be relaxed in applications where sample age or generation time is unknown.

In the original study of Paris et al. (2019a), N was assumed to be known and the likelihood was optimized over s (see below) to get the Maximum Likelihood Estimator (MLE) of selection intensity at the studied locus. One novelty in SelNeTime is to also estimate N by cumulating

data from L loci. Indeed, N reflects the demography of the population so a single value applies to the whole genome. Denoting $\mathbf{y}_l = (y_{l,1}, \ldots, y_{l,n})$ the sequence of observed reference allele counts at locus l, the likelihood of N is evaluated as

$$L(\mathbf{y}_1, \ldots, \mathbf{y}_L; N) = \prod_l P(\mathbf{Y}_l = \mathbf{y}_l | N, s_l = 0)$$

This equation is actually a composite likelihood, as it assumes that the loci are independent (which almost holds in practice if these loci are taken sufficiently far away from each other, but see the discussion). Composite likelihoods are widely used in population genetics and are theoretically unbiased (Wiuf, 2006), although they may result in too narrow (resp. too large) likelihood surfaces when loci are positively (resp. negatively) correlated. Another important assumption in the above equation is that the effect of selection at each locus can be neglected in first approximation while estimating N, i.e. the likelihood of N at each locus l is computed assuming $s_l = 0$; the impact of this approximation on estimation accuracy will be explored through simulations in the manuscript.

To obtain the MLE in practice, we first evaluate the likelihood on a broad grid of candidate N values ranging from 10 to 1,000,000, and next perform a refined search on a second grid of candidate N values centered around the value maximizing the likelihood in the first grid. The likelihood profile obtained from this refined search provides the MLE as well as a 95% confidence interval for N. Importantly, note that only integer values of N are explored during this two step search (besides the biological realism, several steps of the computations require integer N values, both for the WF and the BwS model).

Once N has been estimated, SelNeTime estimates the selection intensity at each locus l based on this value using the same approach as Paris et al. (2019a), i.e. optimizing $P(\mathbf{Y}_l | \hat{N}, s_l)$ over different values of s_l in order to find the MLE (and confidence interval) of this parameter at each locus. As indicated in the notation, s_l is a locus specific parameter, because selective contraints typically differ between loci.

Important features

Estimation of effective population size under neutrality

Using computer simulations, we checked the accuracy of SelNeTime estimation of N, a task that has not been investigated by Paris et al. (2019a). We first considered simulation scenarios without selection, in order to evaluate the method in the favorable situation where its assumptions are met (except that simulations were performed under the true WF model, not the BwS approximation). Estimations of N in the presence of selection are investigated in a later section. As a comparison, we also estimated N for the same data with the method of Hui & Burt (2015), implemented in the NB package (Hui, 2014), which is based on a continuous Beta approximation of allele frequency transitions. Interestingly, both methods performed equally well for short sampling time intervals ($\Delta t = 1$) but the accuracy of NB was reduced for larger time intervals ($\Delta t = 10$) (Figure 1); this is consistent with the higher

probability of fixation events, which are not accounted for in the method of Hui & Burt (2015), when Δt increases, illustrating the advantage of the BwS approximation.

We also explored neutral scenarios with different values of N and of the initial allele frequencies, in order to evaluate the influence of these parameters on SelNeTime estimations. SelNeTime performed almost equally well for all choices of the initial allele frequencies, with a slight advantage for trajectories starting from $x_0 = 0.5$ (Figure S1). This can be explained by the higher probability of fixation events when the initial allele frequency is close to 0 or 1. Indeed, while these events are accounted for in our method, the effect of approximating the true (simulated) WF distribution by the BwS distribution is stronger when the probability of fixation events increases. Estimations of N were also less accurate (in particular they showed a higher variance) for large populations, as expected given the more subtle effect of genetic drift in such scenarios. However, results obtained with $N = 10,000$ (the largest value tested here) remain very good. They could also be improved by increasing the sample size at each time point, which would reduce the sampling variance and thus make the drift variance easier to capture.

Estimation of selection intensity at a locus using the true value of N

We next evaluated the accuracy of SelNeTime to estimate s_l at each locus, assuming that N is known (i.e. that it has been correctly estimated). As this method is essentially a re-implementation of the BwS approach described by Paris et al. (2019a), we first compared it with the inital implementation published by these authors on a limited number of scenarios (Paris et al., 2019b). As expected, we found very similar results between the two implementations (Figure S2), which both lead to almost unbiased estimations of the selection intensity.

To provide a first hint of SelNeTime performance in more diverse scenarios, we then ran it on datasets simulated with various population sizes, selection coefficients and sampling times, with the true value of N still provided (Figure 2). Results showed a larger estimation variance for lower values of N and Δt, but the mean of estimated values was generally very close to the true value of s. A more detailed investigation of the estimation accuracy and of the power to detect selection can be found in the study of Paris et al. (2019a), which explored these questions for a large range of parameter values and sampling designs.

Estimation of effective population size and selection intensity when both are unknown

Finally, we considered the more realistic situation where some loci are under selection and both N and the s_l need to be estimated. In such situations, we remind that SelNeTime would first estimate N, assuming that all loci are neutral, and then estimate s_l at each locus based on the obtained value of N. To evaluate the accuracy of this approach, we first simulated favorable scenarios where only a minor fraction of the genome (here 1%) is under selection, following a classical assumption in population genetics. As shown in Table 1, estimations of N remain very accurate under these conditions, unless when both N and s_l are high where a

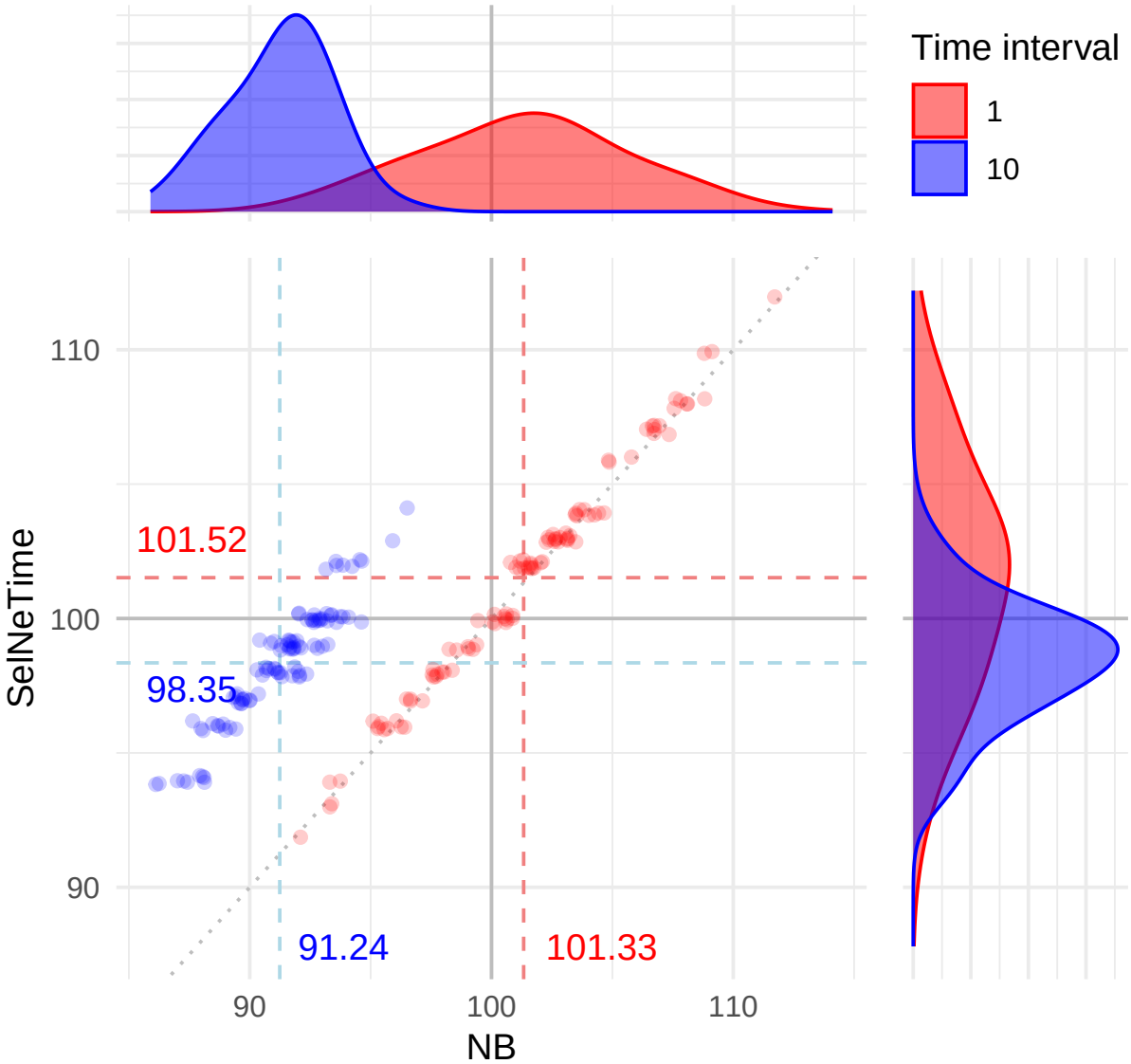

Figure 1: Estimation of the effective population size for 100 datasets, using SelNeTime or NB (Hui, 2014). Each dataset includes 1000 loci and 10 sampling times $t = 1 \ldots 10$ (red) or $t = 10, \ldots, 100$ (blue). Allele frequency trajectories at each locus were simulated independently under the Wright-Fisher model with $N = 100$, $s = 0$ and an initial allele frequency $x_0 = 0.5$. At each time point t_i, sample size is $n_i = 30$.

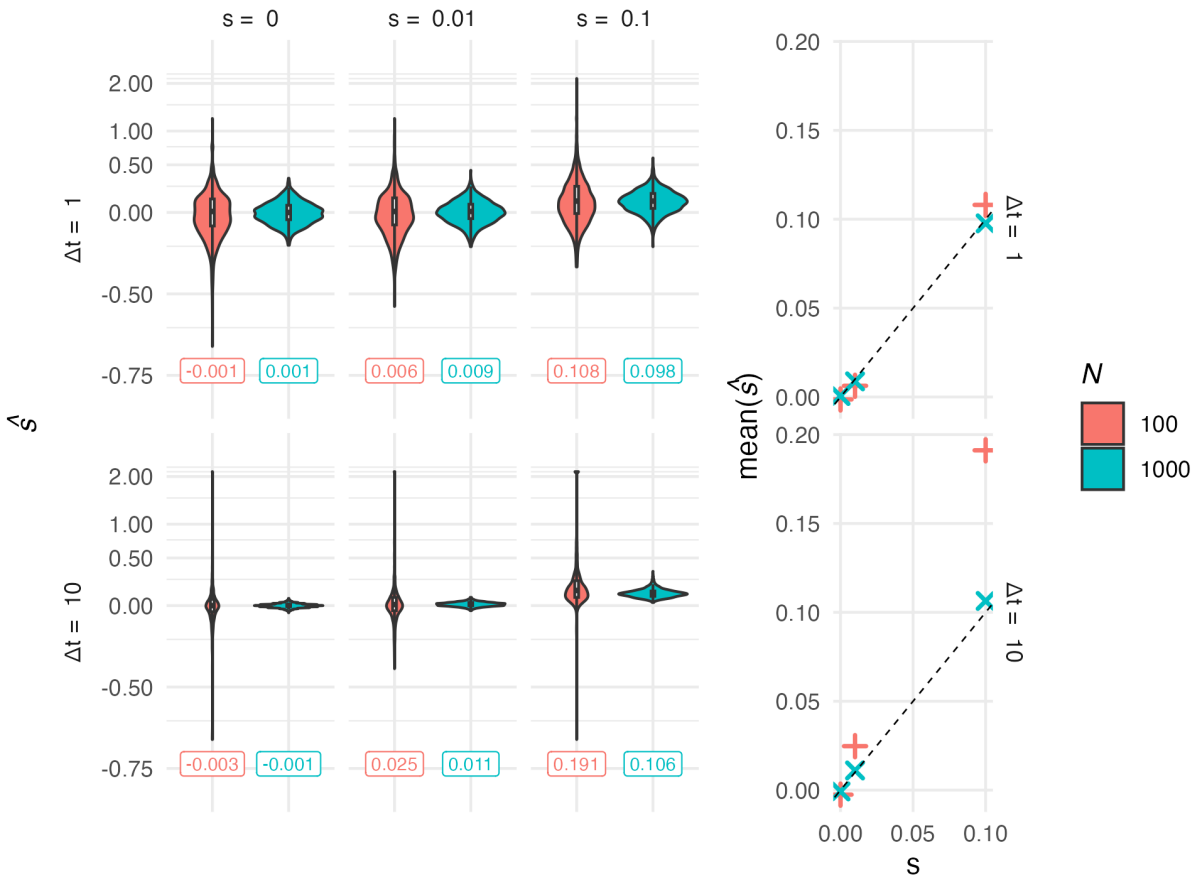

Figure 2: Estimation of the selection intensity using SelNeTime with the true value of N. For given values of s (columns) and N (colors), 1000 loci were considered. Allele frequency trajectories at each locus were simulated independently under the Wright-Fisher model, with $n = 30$ at each of 10 sampling times $t = 1, \ldots, 10$ or $t = 10, \ldots, 100$ (lines), and an initial allele frequency $x_0 = 0.5$. Distribution of estimations for the 100 loci. The mean value for each simulation scenario is shown below the associated violin plot, and on the right part of the plot. A $\log_2(1 + x)$ transformation was applied to the y axis of the left panel.

Table 1: Estimation of N and s for datasets simulated with a proportion $p = 0.01$ of loci under selection. We generated 100 replicates of datasets with 1000 loci and $n = 30$ at each of 10 sampling times $t = 10, \ldots, 100$. Each row shows the estimations means and the median and coverage of 95% confidence intervals for: N (1st row), s_ℓ of loci under selection (2nd row) and s^*_ℓ of neutral loci (3rd row).

		$N = 100$		$N = 1000$	
		$s = 0.01$	$s = 0.1$	$s = 0.01$	$s = 0.1$
$\hat{N}$	mean MLE	98.35	98.06	1015.78	938.48
	median CI	96 — 102	96 — 102	932 — 1106	866 — 1015
	CI coverage	86%	81%	91%	66%
$\hat{s_\ell}$	mean MLE	0.0231	0.2018	0.0106	0.1059
	median CI	-0.076 — 0.112	-0.034 — 0.373	-0.017 — 0.035	0.044 — 0.187
	CI coverage	88.1%	92.7%	91.3%	93.5%
$\hat{s^*_\ell}$	mean MLE	-0.0003	-0.0003	0	0
	median CI	-0.092 — 0.102	-0.092 — 0.102	-0.026 — 0.026	-0.026 — 0.026
	CI coverage	90.6%	90.7%	94.8%	95.4%

small negative bias is observed. Mean estimations of s_l were also accurate in most scenarios, including those where $\hat{N}$ was biased. Note that confidence intervals of s_l were quite large, which is not surprising given that they are locus specific: they are thus based on a single trajectory (in contrast to N).

To investigate whether these good properties would still hold in more challenging scenarios, we next considered increasing proportions of loci under selection, up to the extreme case where all simulated loci are under selection. This generally led to a negative bias in N estimations, which increased with selection intensity and the proportion of the genome under selection (Figure 3). The effect was also more pronounced for larger populations, for instance in the worst case where all loci were selected with $s = 0.1$ the relative bias was about 25% for $N = 100$ and 75% for $N = 1000$. This last observation likely results from the fact that selection is more efficient in large populations; to phrase it in a different way, the effect of selection on genetic diversity is known to be better captured by the product Ns than by s alone (Paris et al., 2019a).

Interestingly, point estimations of s_l were not affected by this (potentially strong) bias in N, even in the most extreme case where all loci were under selection (Table S1, Figure S3). Only confidence intervals of s were affected and appeared to be larger when the negative bias of N was more pronounced. These properties directly result from the fact that s_l is related to the trend of the allele frequency trajectory, while N is related to its variance. An intuitive way to think about it is that providing to SelNeTime a too low value of N does not prevent it to capture the correct trend of the trajectory, but leads it to over-estimate its expected variance and to return a too large confidence interval for this trend.

Having established that the main issue for practical applications of SelNeTime concerns the

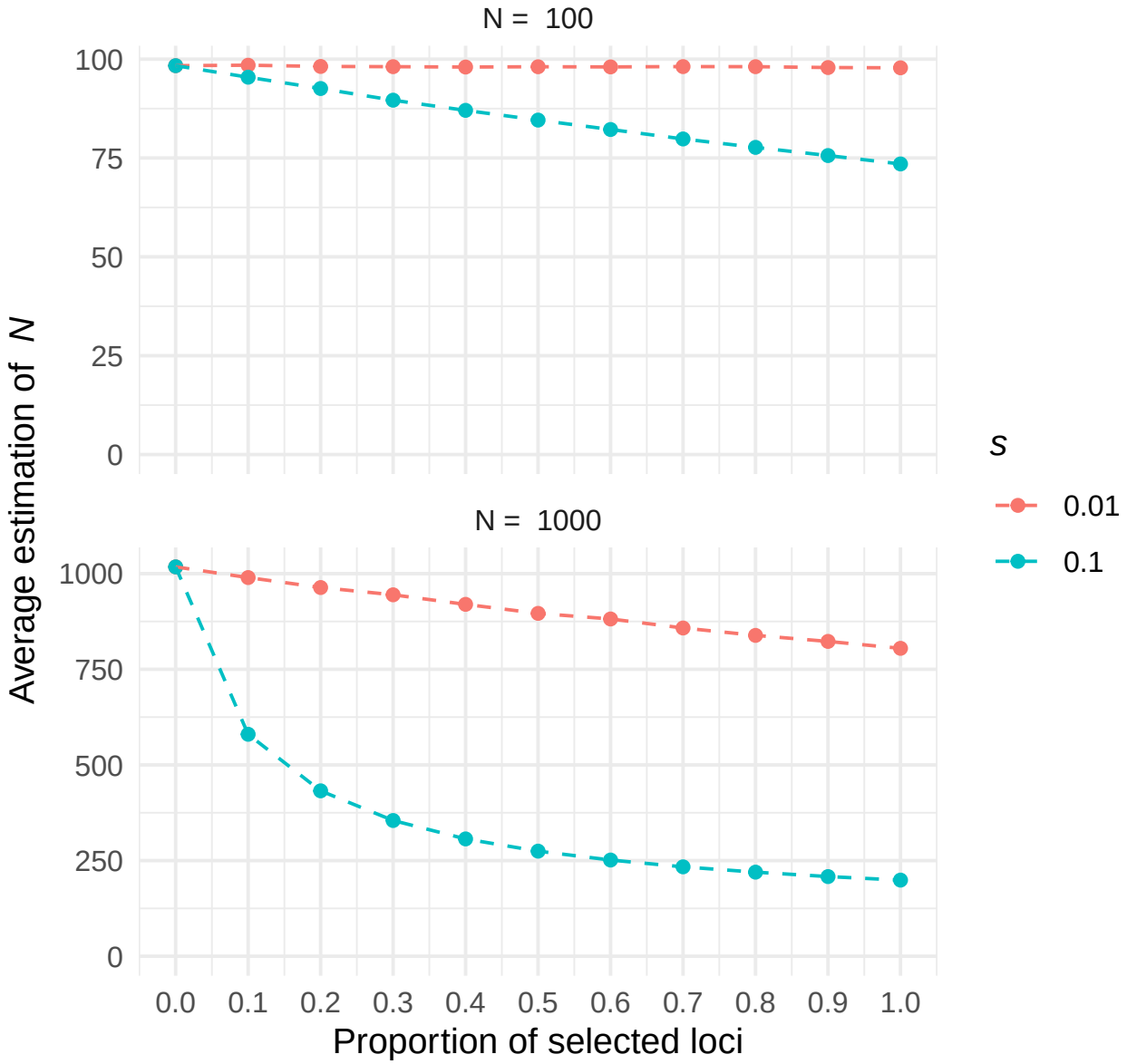

Figure 3: Estimation of the effective population size using SelNeTime, for simulated datasets with various proportions of loci actually under selection (x axis). For a given proportion p, 100 datasets of $1000 * p$ selected loci and $(1-p) * 1000$ neutral loci were considered. Allele frequency trajectories at each locus were simulated independently under the Wright-Fisher model, with $n = 30$ at each of 10 sampling times $t = 10, \ldots, 100$. Initial allele frequencies were set to $x_0 = 0.5$ and the true population size was $N = 100$ (top) or $N = 1000$ (bottom). The average estimated value over the 100 datasets is shown on the y-axis, for different selection intensities (colors)

estimation of N in the presence of selected sites, we finally studied whether this bias could potentially be avoided by accounting for the presence of selection. To do so, we estimated N on datasets where all loci were simulated under selection with the same intensity s_T, but this time providing this true value to SelNeTime (i.e. computing the likelihood given $s = s_T$ instead of $s = 0$). Results were much more accurate, as shown in Figure S4. Mean estimated values with and without the $s = 0$ assumption are summarized in Table 2.

Table 2: Mean estimated values of N over 100 replicates, for datasets simulated with different combinations of N and s_T and a sampling time interval $\Delta t = 10$. Estimations assuming either $s = 0$ or $s = s_T$ are compared.

	$N_T = 100$ $s = 0$	$N_T = 100$ $s = s_T$	$N_T = 1000$ $s = 0$	$N_T = 1000$ $s = s_T$
$s_T = 0$	98.4	–	1018	–
$s_T = 0.01$	97.8	98.4	805	1014
$s_T = 0.1$	73.5	98.5	199	1002

Improved computation time

One important improvement of SelNeTime compared to the inital implementation of the BwS approach in compareHMM (Paris et al., 2019b) concerns the computation time, which was reduced using two main actions. First, the continuous density of the distribution is discretized when building the transition matrices (with a controlled bin width), allowing this continuous part of the distribution to be analyzed exactly in the same way as the discrete probability mass points in 0 and 1 (see section 'The Beta with Spikes approximation'). Second, matrix computations performed at each iteration of the HMM forward algorithm were vectorized, taking advantage of the Numpy library (Harris et al., 2020) to efficiently handle different loci or parameter values in a single operation.

As an illustration, Table 3 shows the computation time required to analyze a time series of 10 samples of 30 observations each, with a sampling time interval of $\Delta t = 10$ generations and a number of loci varying from 100 to 10000.

Table 3: Computation times of CompareHMM (Paris et al., 2019b) and SelNeTime, run on a single CPU of an Intel Xeon $16 * 2.60$ GHz processor with 32Gb of RAM

Nb. loci	compareHMM, estimation of s	SelNeTime, estimation of s	SelNeTime, estimation of N
100	39.27s	28.7s	6.12s
1000	360.01s	36.21s	14.23s
10000	3530.47s	96.95s	87.83s

For 100 loci, SelNeTime runs in approximately the same time as compareHMM, but the

relative performance of SelNeTime improves quickly when the number of loci increases. Actually, the time to compute all transition matrices, which is the first step of both SelNeTime and compareHMM analyses, is independent of the number of loci (but it depends on the sampling design). This part of the analysis took approximately 28s for SelNeTime in this example, which was of the same order as compareHMM, but after this the time spent per locus was far lower with SelNeTime (approx. 0.007s) than with compareHMM (approx. 0.35s).

Comparison with the Wright-Fisher model

While SelNeTime computes by default HMM transitions using the BwS distribution, it might occasionally be useful to compute them based on the WF model, in particular to evaluate the impact of the BwS approximation on inference quality. Although this is numerically possible only for small population sizes (not more than a few hundreds), we included one option to do this in the program. For a true population size of $N = 100$, we could for instance check that using the BwS or the WF model had very limited impact on the inference of either effective population size or selection (Figure S5). A more detailed comparison between the estimations obtained by the BwS and WF distributions can be found in Paris et al. (2019a), but only for selection inference.

Simulation of datasets

As already observed in Paris et al. (2019a) and several other studies on the subject, the estimation accuracy of effective population size or selection depends on both the true evolutionary parameters of the population under study (N and s) and the sampling design of observed data (number and size of observed samples, time intervals between samples). In order to help researchers plan future experiments, or interpret the results obtained for a specific dataset, SelNeTime includes a simulator of observed allele frequency trajectories. Simulations can easily be tunned using either a yaml file or command line options and output data at a format that can directly be analyzed with other SelNeTime commands. Notably, this simulator was used to generate the data underlying the figures and tables of this study.

Discussion

The SelNeTime python package provides a re-implementation of the BwS approach proposed by Paris et al. (2019a) to analyse genomic time series data. Compared to the initial implementation of these authors (Paris et al., 2019b), SelNeTime (i) estimates the effective population size during the temporal period covered by the samples, (ii) runs much faster, epecially for large SNP datasets, and (iii) provides a user-friendly interface with simple commands, including a simulator.

In a first set of simulation scenarios, we validated the fact that SelNeTime accurately infers effective population size in the absence of selection, or selection intensity at each locus assuming the true value of N is provided. Obviously, these two hypothetical conditions are almost never met in practice: populations are subject to selection constraints and their

effective population size is generally unknown. However, what is much less clear is the proportion of the genome that actually evolves as under selection. On one hand, because a large proportion of the genome is non functional and even functional sites do not necessarily impact fitness related traits, one might expect this proportion to be small. On the other hand, accounting for negative (rather than only adaptive/positive) selection and for indirect selection effects due to genetic linkage might substantially increase this proportion and recent empirical studies suggested that linked selection would actually be pervasive in the genome of species like humans (Pouyet et al. (2018)) or drosophila (Elyashiv et al. (2016)). In a second set of simulation scenarios, we thus explored the performance of SelNeTime when both N and the s_l must be estimated, considering various proportions of the genome under selection.

If selection is relatively rare in the genome, SelNeTime was found to accurately infer both N and the s_l (Table 1). For higher proportions of selected sites, N was always under-estimated, but the extent of this bias depended on the true values of both N and s and was for instance limited for small populations or weak selection (Figure 3). Importantly, selection intensity was always accurately inferred, whatever the bias observed for $\hat{N}$ (Table S1). This results is consistent with several recent studies using genomic time series, which suggested that accurate estimations of selection could be achieved despite demographic misspecification (Mathieson & Terhorst, 2022; Whitehouse & Schrider, 2023). Overall, these results suggest that SelNeTime will provide a good estimation of the parameters of interest in a large set of practical situations. Indeed, the original compareHMM approach (Paris et al., 2019b), which is very close to SelNeTime for the estimation of selection parameters, has already been applied to several empirical datasets and provided relevant biological insights (Paris et al., 2019a; Boitard et al., 2021, 2023).

Because the estimation of N might still be an issue when a large part of the genome is affected by selection, a main objective of future research will be to extend the current approach in order to really co-estimate demography and selection. While this is clearly a challenging task, which very few population genetics studies tried to tackle so far, we point out that SelNeTime estimation of N is also accurate in the presence of selection, provided the true value of selection intensity is given as input (Figure S4). Encouragingly, this result suggests that optimization approaches accounting for the effects of both N and the s_l's on the likelihood, could perform well.

Another assumption made by SelNeTime is to consider that allele frequency trajectories observed at distinct loci are independent, an assumption that does not hold in practice for at least two reasons: (i) linkage disequilibrium (LD) and (ii) the so-called Bulmer effect (Bulmer, 1971), i.e. the negative correlation of allele frequencies resulting from the polygenic selection of quantitative traits. While the LD effect can easily be mitigated by sub-sampling SNPs, the Bulmer effect would be more tricky to model with our approach. One first step to explore this question could be to apply SelNeTime to genomic time series simulated under polygenic selection scenarios. In the longer term, we could also try to estimate temporal variations of s_l at each locus (see Mathieson & Terhorst (2022)), which would be more consistent with a model of selection on quantitative traits. Importantly, based on theoretical properties of composite likelihoods, we do not expect that SelNeTime estimates of N will be biased in the

presence of correlations between loci; only the confidence intervals, such as those shown in Figure S5 and returned in the output of the software, may be affected.

Finally, admixture might be another important counfouding effect for our approach, as recently outlined for another method analyzing genomic time series under the assumption of panmixia (Simon & Coop, 2024). The impact of admixture on SelNeTime performance will likely depend on the exact admixture scenario (admixture proportions, single versus recurrent events, overlap of these events with the sampling period, divergence with the source population, local versus global adaptation . . .) so a first step could again be to evaluate it using simulations. It should also be feasible to modify SelNeTime HMM transitions in order to account for specific well documented admixture scenarios, in a similar spirit of what was done in Simon & Coop (2024).

Data, script and code availability

SelNeTime is available on the python package index https://pypi.org/project/selnetime and on the project git repository https://forge.inrae.fr/genetic-time-series/selnetime. All figures of this study can be reproduced using a snakemake pipeline available at https://forge.inrae.fr/genetic-time-series/snt-simulation-pipeline.

Funding

This study has received the support of the Occitanie Regional Council's program 'Key challenge BiodivOc'. Paul Bunel has benefited from State's aid managed by the Agence Nationale de la Recherche under the France 2030 programme, reference ANR-22-PEAE-0005 ("AgroDiv" project of the PEPR Agroécologie et Numérique). He has also been funded by the ECODIV department of INRAE. We are grateful to the genotoul bioinformatics platform Toulouse Occitanie (Bioinfo Genotoul, https://doi.org/10.15454/1.5572369328961167E12) for providing computing resources.

Acknowledgments

A preprint version of this article has been peer-reviewed and recommended by PCI Math Comp Biol (https://doi.org/10.24072/pci.mcb.100406).

We would like to thank Alan Rogers (recommender), Rui Borges (reviewer) and one anonymous reviewer for their constructive comments on the manuscript.

Conflict of interest disclosure

The authors declare they have no conflict of interest relating to the content of this article. SB is a recommender for PCI Math Comp Biol; MN is a recommender for PCI Evol Biol.

References

Boitard S, Liaubet L, Paris C, Fève K, Dehais P, Bouquet A, Riquet J, Mercat M-J (2023) Whole-genome sequencing of cryopreserved resources from french large white pigs at two distinct sampling times reveals strong signatures of convergent and divergent selection between the dam and sire lines. *Genetics Selection Evolution*, **55**, 13.

Boitard S, Paris C, Sevane N, Servin B, Bazi-Kabbaj K, Dunner S (2021) Gene banks as reservoirs to detect recent selection: The example of the asturiana de los valles bovine breed. *Frontiers in genetics*, **12**.

Bollback JP, York TL, Nielsen R (2008) Estimation of $2N_e s$ from temporal allele frequency data. *Genetics*.

Bulmer M (1971) The effect of selection on genetic variability. *The American Naturalist*, **105**, 201–211.

Elyashiv E, Sattath S, Hu TT, Strutsovsky A, McVicker G, Andolfatto P, Coop G, Sella G (2016) A genomic map of the effects of linked selection in drosophila. *PLoS genetics*, **12**, e1006130.

Fages Antoine, Hanghøj Kristian, Khan Naveed, Gaunitz Charleen, Seguin-Orlando Andaine, Leonardi Michela, McCrory Constantz Christian, Gamba Cristina, Al-Rasheid Khaled A.S., Albizuri Silvia, Alfarhan Ahmed H., Allentoft Morten, Alquraishi Saleh, Anthony David, Baimukhanov Nurbol, Barrett James H., Bayarsaikhan Jamsranjav, Benecke Norbert, Bernáldez-Sánchez Eloísa, Berrocal-Rangel Luis, Biglari Fereidoun, Boessenkool Sanne, Boldgiv Bazartseren, Brem Gottfried, Brown Dorcas, Burger Joachim, Crubézy Eric, Daugnora Linas, Davoudi Hossein, Peter de BD, María de los Á de Cy de V-C, Deschler-Erb Sabine, Detry Cleia, Dill Nadine, Maria do MO, Dohr Anna, Ellingvåg Sturla, Erdenebaatar Diimaajav, Fathi Homa, Felkel Sabine, Fernández-Rodríguez Carlos, García-Viñas Esteban, Germonpré Mietje, Granado José D., Hallsson Jón H., Hemmer Helmut, Hofreiter Michael, Kasparov Aleksei, Khasanov Mutalib, Khazaeli Roya, Kosintsev Pavel, Kristiansen Kristian, Kubatbek Tabaldiev, Kuderna Lukas, Kuznetsov Pavel, Laleh Haeedeh, Leonard Jennifer A., Lhuillier Johanna, Liesau von Lettow-Vorbeck Corina, Logvin Andrey, Lõugas Lembi, Ludwig Arne, Luis Cristina, Arruda Ana Margarida, Marques-Bonet Tomas, Matoso Silva Raquel, Merz Victor, Mijiddorj Enkhbayar, Miller Bryan K., Monchalov Oleg, Mohaseb Fatemeh A., Morales Arturo, Nieto-Espinet Ariadna, Nistelberger Heidi, Onar Vedat, Pálsdóttir Albína H., Pitulko Vladimir, Pitskhelauri Konstantin, Pruvost Mélanie, Rajic Sikanjic Petra, Rapan Papeša Anita, Roslyakova Natalia, Sardari Alireza, Sauer Eberhard, Schafberg Renate, Scheu Amelie, Schibler Jörg, Schlumbaum Angela, Serrand Nathalie, Serres-Armero Aitor, Shapiro Beth, Sheikhi Seno Shiva, Shevnina Irina, Shidrang Sonia, Southon John, Star Bastiaan, Sykes Naomi, Taheri Kamal, Taylor William, Teegen Wolf-Rüdiger, Trbojević Vukičević Tajana, Trixl Simon, Tumen Dashzeveg, Undrakhbold Sainbileg, Usmanova Emma, Vahdati Ali, Valenzuela-Lamas Silvia, Viegas Catarina, Wallner Barbara, Weinstock Jaco, Zaibert Victor, Clavel Benoit, Lepetz Sébastien, Mashkour Marjan, Helgason Agnar, Stefánsson Kári, Barrey Eric, Willerslev Eske, Outram Alan K., Librado Pablo, Orlando Ludovic (2019) Tracking Five Millennia of Horse Management with Extensive Ancient Genome Time Series. *Cell*, **177**, 1419–1435.e31.

Harris CR, Millman KJ, Walt SJ van der, Gommers R, Virtanen P, Cournapeau D, Wieser E, Taylor J, Berg S, Smith NJ, Kern R, Picus M, Hoyer S, Kerkwijk MH van, Brett M, Haldane A, Río JF del, Wiebe M, Peterson P, Gérard-Marchant P, Sheppard K, Reddy T, Weckesser W, Abbasi H, Gohlke C, Oliphant TE (2020) Array programming with NumPy. *Nature*, **585**, 357–362. https://doi.org/10.1038/s41586-020-2649-2

Hui T-YJ (2014) NB: Maximum likelihood method in estimating effective population size from genetic data. *CRAN repository*.

Hui T-YJ, Burt A (2015) Estimating Effective Population Size from Temporally Spaced Samples with a Novel, Efficient Maximum-Likelihood Algorithm. *Genetics*, **200**, 285–293.

Lacerda M, Seoighe C (2014) Population genetics inference for longitudinally-sampled mutants under strong selection. *Genetics*, **198**, 1237–1250.

Long A, Liti G, Luptak A, Tenaillon O (2015) Elucidating the molecular architecture of adaptation via evolve and resequence experiments. *Nature Reviews Genetics*, **16**, 567–582.

Mathieson I, Terhorst J (2022) Direct detection of natural selection in bronze age britain. *Genome Research*, **32**, 2057–2067.

Paris C, Servin B, Boitard S (2019b) compareHMM. *GitHub repository*.

Paris C, Servin B, Boitard S (2019a) Inference of selection from genetic time series using various parametric approximations to the wright-fisher model. *G3: Genes, Genomes, Genetics*, **9**, 4073–4086.

Pouyet F, Aeschbacher S, Thiéry A, Excoffier L (2018) Background selection and biased gene conversion affect more than 95% of the human genome and bias demographic inferences. *Elife*, **7**, e36317.

Simon A, Coop G (2024) The contribution of gene flow, selection, and genetic drift to five thousand years of human allele frequency change. *Proceedings of the National Academy of Sciences*, **121**, e2312377121.

Tataru P, Bataillon T, Hobolth A (2015) Inference under a wright-fisher model using an accurate beta approximation. *Genetics*, **201**, 1133–1141. https://doi.org/10.1534/genetics.115.179606

Uhl M, Bunel P, Navascués M de, Boitard S, Servin B (2025) Snt. *Forge INRAE repository*.

Whitehouse LS, Schrider DR (2023) Timesweeper: Accurately identifying selective sweeps using population genomic time series. *Genetics*, **224**, iyad084.

Wiuf C (2006) Consistency of estimators of population scaled parameters using composite likelihood. *Journal of mathematical biology*, **53**, 821–841.

Supplementary material

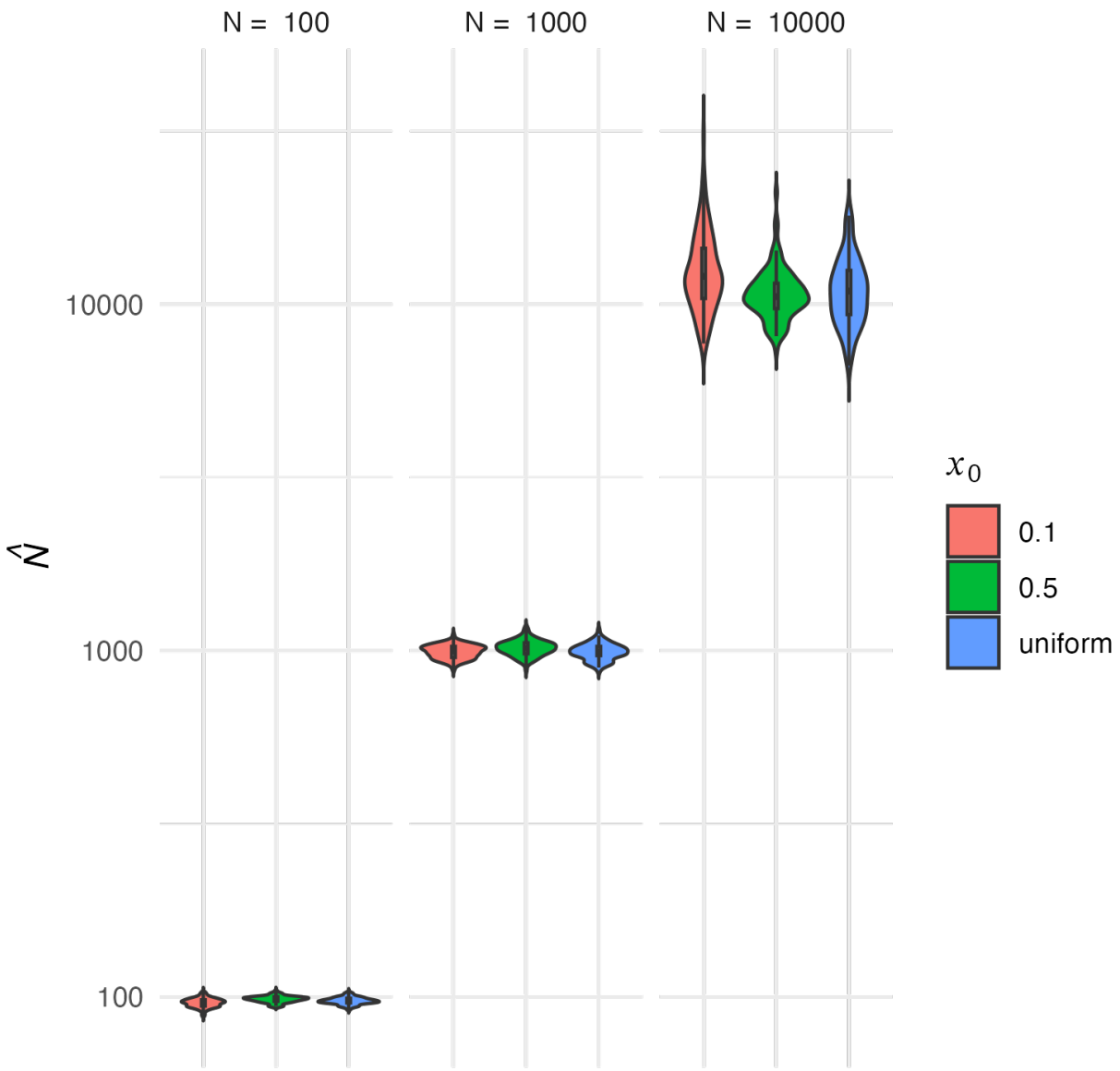

Figure S1: Estimation of the effective population size using SelNeTime, for different values of N (panels) and different values / distributions of the initial allele frequency x_0 (colors). For each combination of these two parameters, 100 datasets of 1000 loci were considered. Allele frequency trajectories at each locus were simulated independently under the Wright-Fisher model, with $n = 30$ at each of 10 sampling times $t = 10, \ldots, 100$.

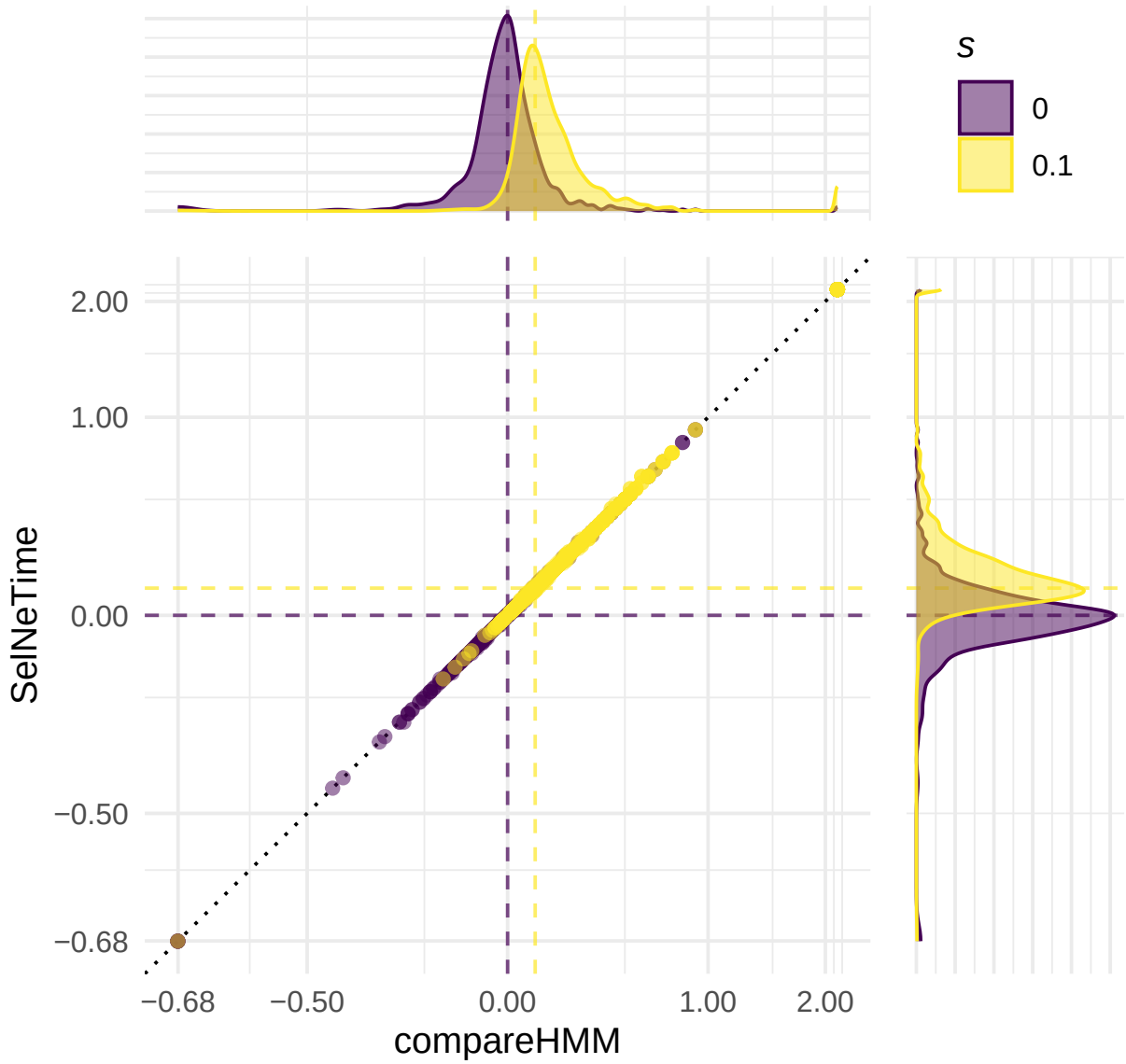

Figure S2: Estimation of the selection intensity for 1000 loci using SelNeTime and compareHMM. Allele frequency trajectories at each locus were simulated independently under the Wright-Fisher model with $N = 100$, $s = 0$ (purple) or $s = 0.1$ (yellow) and an initial allele frequency $x_0 = 0.5$. Allele frequencies were observed for $n = 30$ at 10 sampling times $t = 10 \ldots 100$. A $\log_2(1 + x)$ transformation was applied to both axes, in order to have a qualitatively symmetrical scale around 0 (a selection of -0.5 is the opposite of $s = 1$, etc.)

		$N = 100$		$N = 1000$	
		$s = 0.01$	$s = 0.1$	$s = 0.01$	$s = 0.1$
$\hat{N}$	mean MLE	97.81	73.5	804.77	198.96
	median CI	96 — 100	71 — 76	743 — 860	192 — 206
	CI coverage	82%	0%	0%	0%
$\hat{s}_\ell$	mean MLE	0.0220	0.1978	0.0109	0.0995
	median CI	-0.076 — 0.122	-0.059 — 0.411	-0.017 — 0.044	-0.009 — 0.236
	CI coverage	89.5%	95.9%	93.5%	99.8%

Table S1: Estimation of N and s for datasets simulated with all loci under selection. We generated 100 replicates of datasets with 1000 loci and $n = 30$ at each of 10 sampling times $t = 10, \ldots, 100$. Each row shows the estimations mean and the median and coverage of 95% confidence intervals for: N (1st row), and s_ℓ of loci under selection (2nd row).

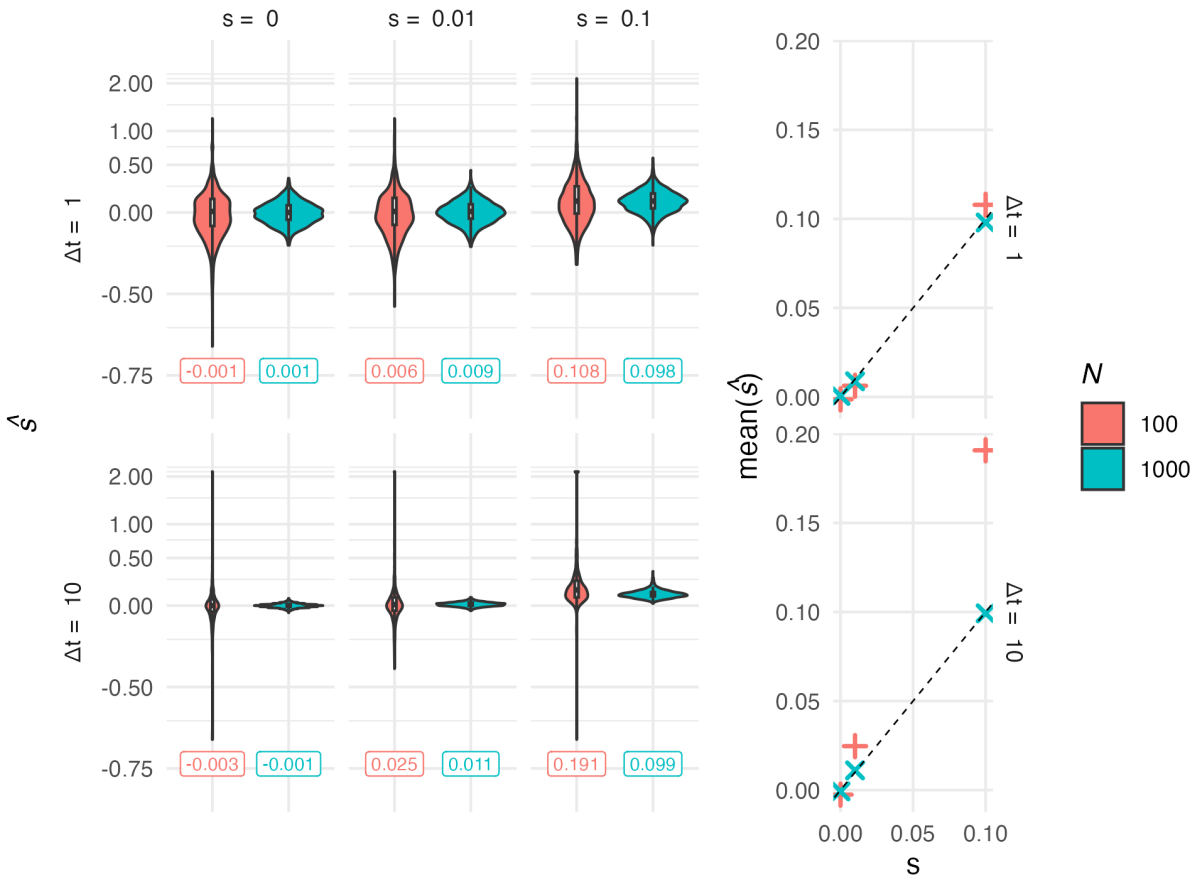

Figure S3: Estimation of the selection intensity using SelNeTime without knowing N. For given values of s (columns) and N (colors), 1000 loci were considered. Allele frequency trajectories at each locus were simulated independently under the Wright-Fisher model, with $n = 30$ at each of 10 sampling times $t = 1, \ldots, 10$ or $t = 10, \ldots, 100$ (lines), and an initial allele frequency $x_0 = 0.5$. Distribution of estimations for the 100 loci. The mean value for each simulation scenario is shown below the associated violin plot, and on the right part of the plot. A $\log_2(1 + x)$ transformation was applied to the y axis of the left panel.

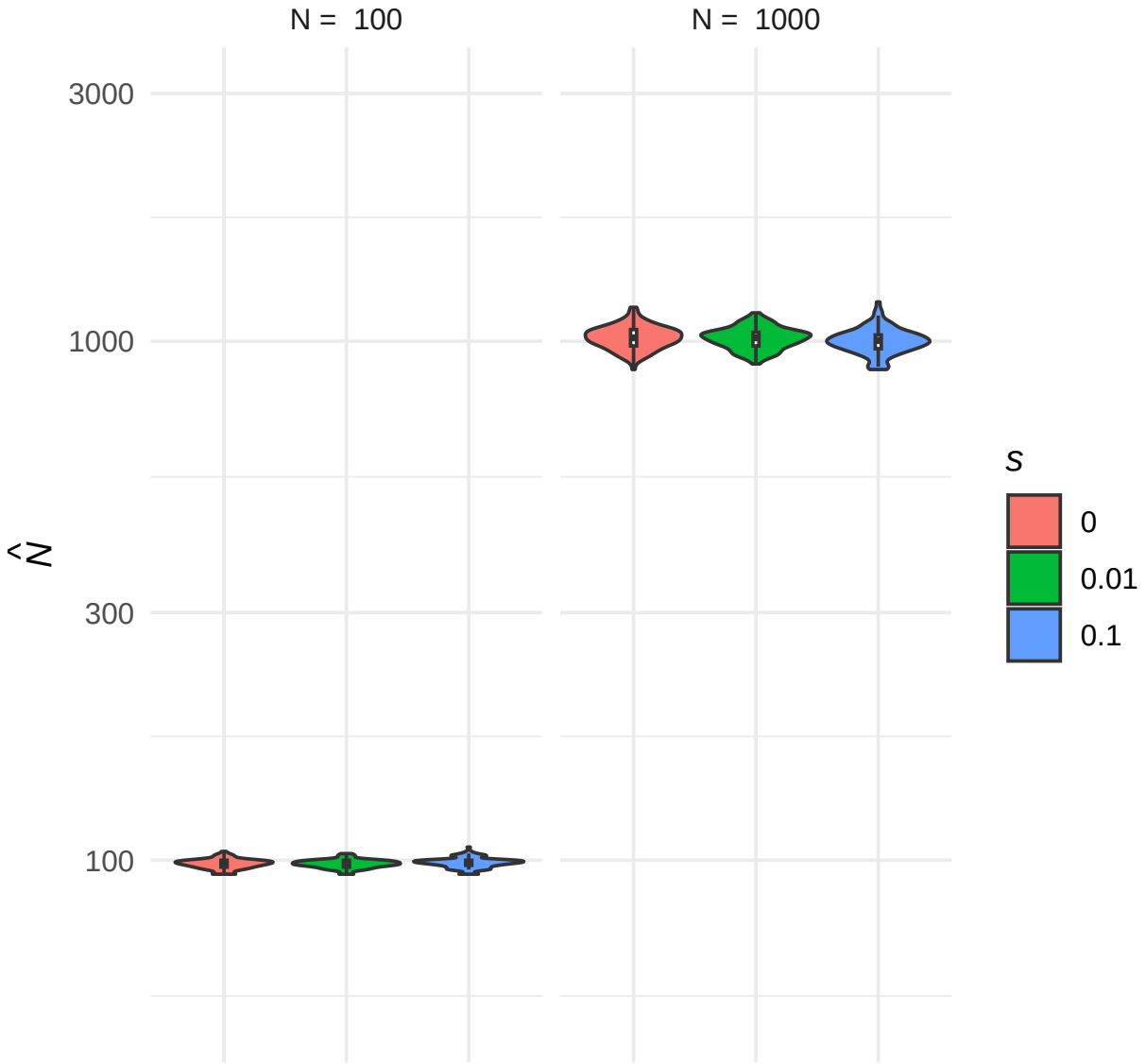

Figure S4: Estimation of the effective population size using SelNeTime with the true value of s provided, for simulated datasets where all loci are under selection with the same intensity. For given values of s (colors) and N (panels) 100 datasets of 1000 loci were considered. Allele frequency trajectories at each locus were simulated independently under the Wright-Fisher model, with $n = 30$ at each of 10 sampling times $t = 10, \ldots, 100$. Initial allele frequencies were set to $x_0 = 0.5$.

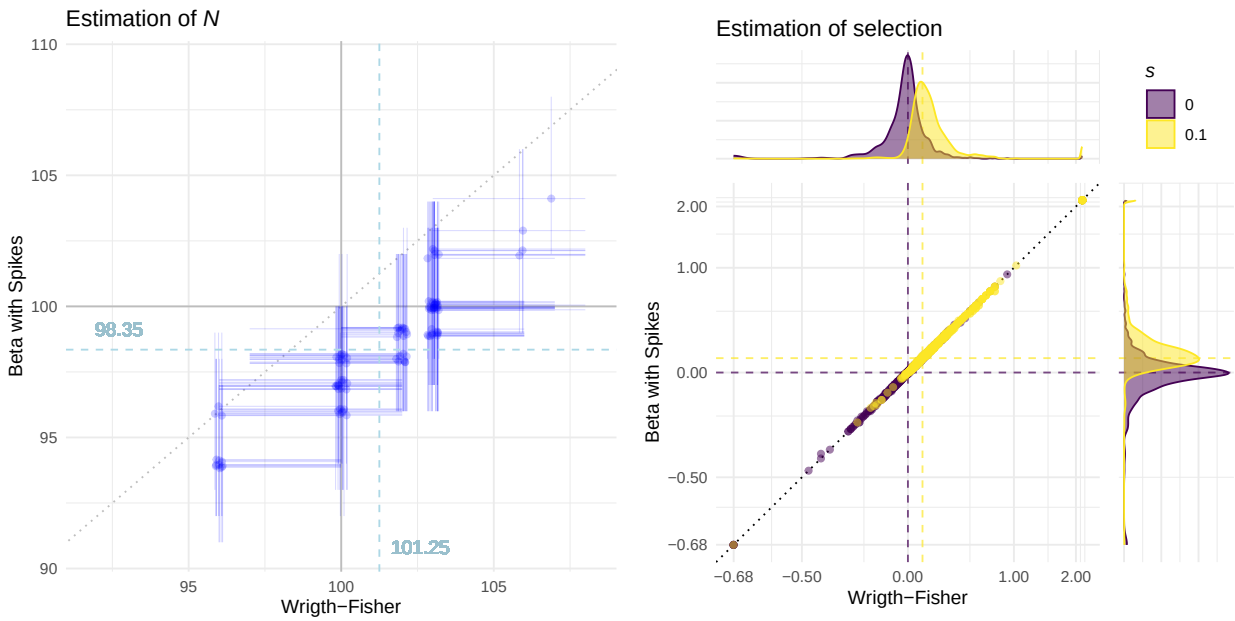

Figure S5: Comparison of SelNeTime estimations based on the Beta with Spikes and the Wright-Fisher model. **(a)**: MLE and 95% confidence interval of effective population size for 100 datasets of 1000 loci each. **(b)** MLE of s for 1000 loci. The two panels show the estimations obtained using the WF (x axis) or the BwS (y axis) model on the same data, which were simulated independently under the Wright-Fisher model with $N = 100$ $s = 0$ (Left and Right, purple) or $s = 0.1$ (Right, yellow) and initial allele frequencies $x_0 = 0.5$. Allele frequencies were observed for $n = 30$ at 10 sampling times $t = 10 \ldots 100$. A $\log_2(1 + x)$ transformation was applied to both axes of panel b in order to restore the symmetry between positive and negative values of s (for instance, $s = -0.5$ implies that the fitness of reference homozygote genotypes is twice that of alternative homozygote genotypes, the symmetric case of $s = 1$).