



Detecting Positive Selection in Populations Using Genetic Data

Angelos Koropoulis, Nikolaos Alachiotis, and Pavlos Pavlidis

Abstract

High-throughput genomic sequencing allows to disentangle the evolutionary forces acting in populations. Among evolutionary forces, positive selection has received a lot of attention because it is related to the adaptation of populations in their environments, both biotic and abiotic. Positive selection, also known as Darwinian selection, occurs when an allele is favored by natural selection. The frequency of the favored allele increases in the population and, due to genetic hitchhiking, neighboring linked variation diminishes, creating so-called selective sweeps. Such a process leaves traces in genomes that can be detected in a future time point. Detecting traces of positive selection in genomes is achieved by searching for signatures introduced by selective sweeps, such as regions of reduced variation, a specific shift of the site frequency spectrum, and particular linkage disequilibrium (LD) patterns in the region. A variety of approaches can be used for detecting selective sweeps, ranging from simple implementations that compute summary statistics to more advanced statistical approaches, e.g., Bayesian approaches, maximum-likelihood-based methods, and machine learning methods. In this chapter, we discuss selective sweep detection methodologies on the basis of their capacity to analyze whole genomes or just subgenomic regions, and on the specific polymorphism patterns they exploit as selective sweep signatures. We also summarize the results of comparisons among five open-source software releases (SweeD, SweepFinder, SweepFinder2, OmegaPlus, and RAiSD) regarding sensitivity, specificity, and execution times. Furthermore, we test and discuss machine learning methods and present a thorough performance analysis. In equilibrium neutral models or mild bottlenecks, most methods are able to detect selective sweeps accurately. Methods and tools that rely on linkage disequilibrium (LD) rather than single SNPs exhibit higher true positive rates than the site frequency spectrum (SFS)-based methods under the model of a single sweep or recurrent hitchhiking. However, their false positive rate is elevated when a misspecified demographic model is used to build the distribution of the statistic under the null hypothesis. Both LD and SFS-based approaches suffer from decreased accuracy on localizing the true target of selection in bottleneck scenarios. Furthermore, we present an extensive analysis of the effects of gene flow on selective sweep detection, a problem that has been understudied in selective sweep literature.

Key words Positive selection, Selective sweep, Software tools, Summary statistics, Machine learning

1 The Selective Sweep Theory

When a strongly beneficial mutation occurs and spreads in a population, the frequency of linked neutral (or weakly negatively selected) variants will increase. In a seminal paper, Smith and

Haigh [86] described this process, for which they coined the term genetic hitchhiking. They showed that in large populations, where random genetic drift is negligible, hitchhiking can drastically reduce genetic variation near the site/locus favored by natural selection. Due to the local reduction of genetic diversity, which is swept by natural selection, the process is called “selective sweep.”

The selective sweep model predicts that in recombining chromosomal regions diversity vanishes at the site of selection immediately after the fixation of the beneficial allele. Due to recombination, genetic diversity is predicted to increase as a function of the distance to the selected site (scaled by the selection coefficient and the recombination rate). As a result, the genetic diversity is maintained due to recombination in genomic regions that are in the proximity of a selective sweep: SNPs are not generated by novel mutations, but they are old mutations that escaped selection because of recombination. This result is also roughly correct in finite populations [49, 90]. Further signatures of the hitchhiking effect include (1) shifts in the SFS of polymorphisms such as an excess of low- and high-frequency derived alleles [22, 37], and (2) an elevated level of LD in the early phase of the fixation process of a beneficial mutation [51, 91]. It is important to note that the aforementioned signatures of a selective sweep are predicted when (1) fixation of the beneficial mutation has just been completed; (2) recombination rate is positive, i.e., the chromosome is recombining; (3) the population size is approximately constant over time; (4) the population is isolated; (5) no gene conversion has occurred in the proximity of the beneficial mutation. Despite the relatively strict assumptions of the selective sweep model, several tests have been developed that exploit the properties of the hitchhiking effect to map recent, strong, positive directional selection along recombining chromosomes of several species.

Searching for strong positive selection in the genomes of individuals of a natural population has been the focus of a multitude of studies over the past years [3, 8, 41, 52, 67, 78, 97, 100, 102]. The goals of these studies have been (1) to provide evidence of positive selection, (2) estimate the strength of selection, and (3) localize the targets of selection. Thus, these studies aim to provide insights into the genetical mechanisms of adaptation either in wild populations or during domestication. A long-term goal is that the genes that experienced recent and strong positive selection could be identified and the associated functions and phenotypes characterized.

Early studies of selective sweep localization followed a two-tier approach: at first, levels of DNA polymorphism were measured for a very large number of loci on a genome-wide scale within populations. The goal of this initial step was to identify loci with reduced diversity compared to divergence with another species. The diversity–divergence contrast highlights regions with reduced intra-population diversity compared to what is expected from the

divergence data. Thus, divergence is treated as a proxy for the mutation rate. Some studies employed microsatellite markers to measure polymorphism and searched for regions of depleted variability as an indicator of a selective sweep due to genetic hitchhiking in the region. In the second step, a thorough sequencing of the candidate regions was performed and a selective sweep detection pipeline was executed. A statistical problem related to this procedure springs from the fact that regions analyzed for the occurrence of a selective sweep do not represent a random fraction of the genome. Instead, they are outliers since they are characterized by decreased amounts of diversity. A proper statistical testing for the hypothesis of a selective sweep requires the null distribution of the statistic to be built from neutral regions with the same properties (e.g., outliers for diversity levels) [93]. With the advent of next generation sequencing, the candidate gene approach is replaced by full genome screenings for positive selection, thus the statistical problem of testing outlier genes for positive selection is diminished at least for the model organisms. For non-model organisms, where a reference genome is still missing a candidate gene approach could provide insights into their adaptation processes.

2 Methods to Detect Selective Sweeps in Genome-Wide Data

2.1 Detecting Sweeps Based on Diversity Reduction

The most striking and persistent effect of genetic hitchhiking is the reduction of diversity. Smith and Haigh [86] predicted the reduction of heterozygosity immediately after the fixation of the beneficial mutation. Especially in genomic regions with reduced recombination rate per physical distance, the reduction of diversity is expected to be evident. Subsequent studies [1, 2, 15, 53, 62, 89, 90] confirmed this prediction for *D. melanogaster*, *D. simulans*, and *D. ananassae* species. Charlesworth et al. [27], however, showed that a similar prediction holds for background selection as well: if neutral variants are linked to a strongly deleterious mutation, the level of polymorphism diminishes while the deleterious mutation is gradually removed from the population. The amount of polymorphism reduction depends on the selection coefficient of the deleterious mutation [35]. For example, for lethal mutations there is no polymorphism reduction effect since it is being directly removed from the population. Innan and Stephan [47] demonstrated that in a hitchhiking model, the estimated level of diversity, $\hat{\theta}$, is negatively correlated with $\hat{\theta}/\rho$, where ρ is the recombination rate. In contrast, in a background selection model, the estimated level of diversity is positively correlated with the same quantity (see also ref. 88 for a review).

2.2 The SFS Signature of a Selective Sweep

The studies by Braverman et al. [22] and Fay and Wu [37] showed that a selective sweep shifts the SFS toward high- and low-frequency derived variants. Neutral variants that are initially linked to the beneficial variant increase in frequency, whereas variants that are initially not linked to the beneficial variant decrease in frequency during the fixation of the beneficial mutation.

A breakthrough on detecting selective sweeps was proposed by Kim and Stephan [52], known as the Kim and Stephan test. They developed a composite-likelihood-ratio (CLR) test to compare the probability of the observed polymorphism data under the standard neutral model with the probability of observing the data under a model of selective sweep. The Kim and Stephan test is a maximum-likelihood-based test that reports the value of $a = 4N_e s$, where s is the selection coefficient that maximizes the CLR. The Kim and Stephan test was the first to implement a CLR test on sweep detection. Due to its inefficient implementation, however, it has been used to detect selection only in candidate loci [16, 80]. Furthermore, it adopts several oversimplified assumptions. First, the neutral model was derived by an equilibrium neutral population, i.e., a population with constant population size. Second, the selection model was derived by Fay and Wu's model [37], where only the low- and the high-frequency derived classes are assumed.

2.3 The LD Signature of a Selective Sweep

The third signature of a selective sweep refers to a specific pattern of LD that emerges in the neighborhood of the beneficial mutation. Upon fixation of the beneficial mutation, elevated levels of LD emerge on each side of the selected site, whereas a decreased LD level is observed between polymorphisms found on different sides of the selected site. The high LD levels on the different sides of the selected locus are due to the fact that *a single* recombination event allows multiple polymorphisms *on the same side of the sweep* to escape the sweep. Between those SNPs the level of LD will be high. On the other hand, polymorphisms that reside on different sides of the selected locus need a minimum of two recombination events, thus LD is decreased. Figure 1 shows an example of the LD patterns emerging after a sweep.

The LD-based signature of a selective sweep was proposed and thoroughly investigated by Kim and Nielsen [51]. In this study, Kim and Nielsen introduced a simple statistic, named ω -*statistic*, that facilitates the detection of the specific LD patterns that emerge after a sweep. For a window of W SNPs that is split into two non-overlapping subregions L and R , with l and $W-l$ SNPs, respectively, the ω -*statistic* is computed as follows:

$$\omega = \frac{\binom{l}{2} + \binom{W-l}{2}^{-1} \left(\sum_{i,j \in L} r_{ij}^2 + \sum_{i,j \in R} r_{ij}^2 \right)}{\left(l(W-l) \right)^{-1} \sum_{i \in L, j \in R} r_{ij}^2}. \quad (1)$$

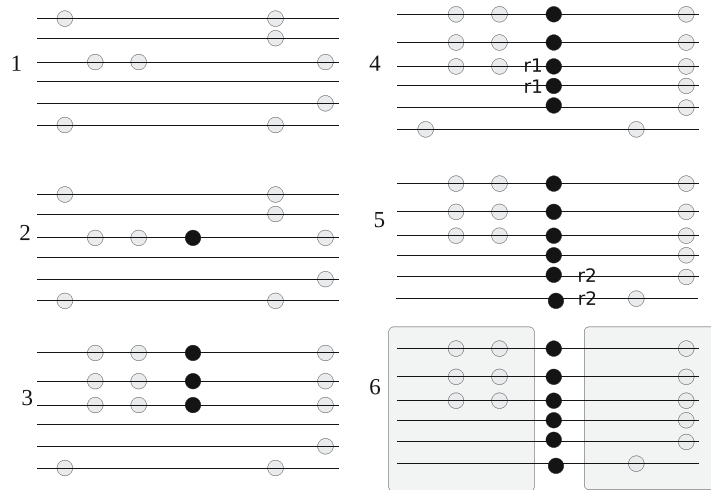


Fig. 1 The LD signature of a complete hard selective sweep. Assume a population with neutral segregating variation (1). A beneficial mutation occurs (shown as a black allele) in subfigure (2). Since the mutation is beneficial, its frequency will increase in the population. Neutral variants that are linked to the beneficial mutation will hitchhike with it (3). Due to recombination, mutations from a neutral background will get linked with the beneficial mutation (4, 5). The recombination events are depicted on the locations of the involved chromosomes by $r1$ and $r2$, respectively. Finally, the selective sweep completes (6). The LD pattern that emerges from such a process is the elevated LD on each side of the beneficial mutation and the decreased LD for SNPs that are on different sides of the beneficial mutation. The figure is adapted from [69]

2.4 Detecting Sweeps Using Machine Learning Methods

The process of detecting genomic regions that have been affected by positive selection can be treated as a classification problem for which each genomic region is classified as either neutral or selected. If the parameters of the selective sweep and the demographic model are known, then disentangling a selective sweep from demography can be treated as a typical binary classification problem. In computer science and mathematics, theoretical and algorithmic advancements have been developed the last decades that perform classification of datasets. These advancements can be grouped as machine learning methods, because first they train computers to understand patterns from the data, and then use this knowledge to classify an unknown sample. Their application in population genetics still remains limited, even though the last years a few methods have been developed [57, 71, 82]. The first application of machine learning in population genetics to our knowledge was developed by Pavlidis et al. [71], who used a support vector machine approach to perform the classification. Pavlidis et al. [71] used as features results from the CLR test (SFS-based) and the ω - *statistic* as well as the difference between the locations that each of the aforementioned tests pinpoint. Lin et al. [57] also developed a machine learning approach based on the “boosting” algorithm, a statistical method

that combines simple classification rules using summary statistics to maximize their joint predictive performance. More recently, Schrider and Kern [82] proposed an extremely randomized trees classifier to identify soft selective sweeps, hard selective sweeps, their linked regions, and neutral regions. Their software is called “S/HIC.” A new version of “S/HIC” (called diploS/HIC) was proposed by [50] that can also use unphased genotypes in contrast to “S/HIC.” The application of machine learning tools in population genetics has been reviewed in [83].

Typically, in a supervised learning problem, the goal is to accurately predict previously unseen data based on a set of already seen data (training data). The problem can be formulated as training the computer to recognize the combinations of feature-values that are associated with either of the classes. Here, the class of each data point is encoded as “neutrality/selection.” In contrast to other disciplines in which machine learning methods are applied, the number of well-annotated examples that the algorithm requires for its training is limited. In fact, all “known” targets of selection do not represent any established truth but are predictions of algorithms that are mostly based on simplistic models. Even though there is a general agreement about the validity of positive selection detection in loci such as the LCT [17], the historical truth, i.e., whether a locus was indeed selected by natural selection remains unknown. Even if we did know the definite true targets of selection, it would still be challenging to build an accurate predictor based on them. The reason is that those training examples would be obtained from heterogeneous populations that have experienced and would incorporate a multitude of other evolutionary forces besides positive selection. A remedy for the aforementioned problems is to use simulated results for the training of the machine learning algorithms. On one hand, simulated data ensure the control of heterogeneity of the training samples as well as the correctness of the assigned class. On the other hand, the simulation process does not capture the whole set of stochastic processes that affect the data. Thus, even though training and evaluation processes perform well on simulated data, they might perform poorly on real data. In this study, we present an extensive testing of machine learning methodologies in Subheading 6.

3 The Problem of Demography

Demography poses severe challenges on the selection detection process due to the fact that it may generate SNP patterns that resemble the signatures of genetic hitchhiking. In recombining chromosomes, selective sweep detection becomes feasible mainly due to two factors: (1) the fixation of the beneficial mutation, and (2) the fact that coalescent events occur at a higher rate in the

presence of a sweep than they do in its absence. It is these two factors, along with *recombination events*, that generate the specific signatures of a selective sweep, enabling us to detect traces of positive selection in genomes. However, additional factors can also trigger a high rate of coalescent events, leading to the generation of similar (to a selective sweep) signatures in the genome, and thus misleading current selective sweep detection approaches. For instance, assume a bottleneck event that is characterized by three phases: (1) a recent phase of large effective population size, (2) a second phase, prior to the first one, of small population size (the bottleneck phase), and (3) an ancestral period of large population size. It is due to the decrease of the effective population size in the bottleneck phase that a high rate of coalescent events occur in a relatively short period of time. Furthermore, lineages can escape the bottleneck, passing to the ancestral phase of large effective population size, and therefore requiring more time to coalesce. In a recombining chromosome, genomic regions that are characterized by short coalescent trees due to massive coalescent events may alternate with genomic regions with lineages that have escaped the bottleneck phase (*see* Fig. 2). Such alternations can generate SNP

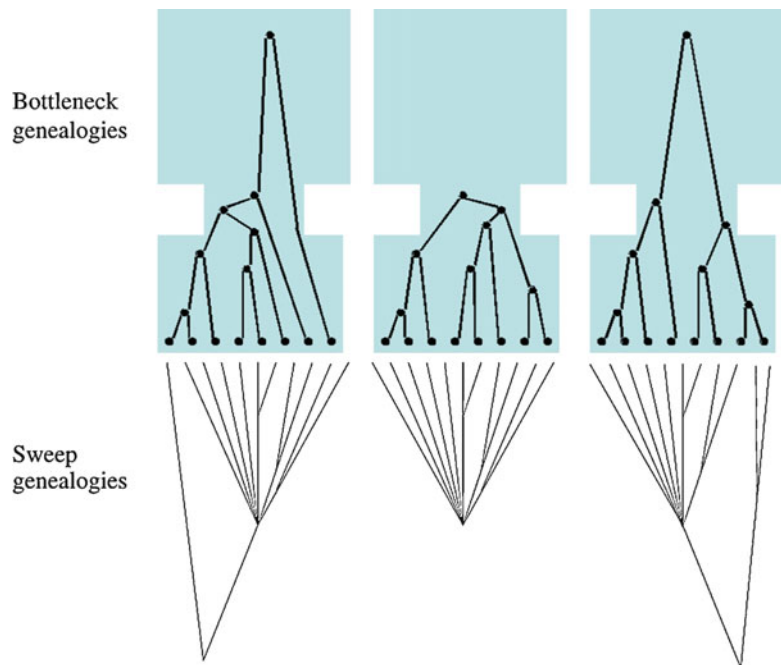


Fig. 2 Bottleneck demographic scenarios (top panel) may result in similar genealogies to a selective sweep (bottom panel). Both models may produce very short coalescent trees. As we move from the selection site, selective sweeps produce genealogies with long internal branches. Similarly, bottlenecks may produce genealogies with very long internal branches if the ancestral population size is large

patterns that are highly similar to those generated by a selective sweep, yielding the detection process very challenging, if not infeasible [70].

Besides demographic bottlenecks, other demographic scenarios may also generate SNP patterns that resemble those of a selective sweep. Recently, Alachiotis and Pavlidis [5] demonstrated that gene flow (migration) between populations poses severe challenges to existing sweep detection methods, suggesting that appropriate sweep signatures for migration models are yet to be found (figure 2 in [5]). Similarly, De and Durrett [31] demonstrated that both the LD and the SFS are affected if a stepping stone spatial structure characterizes the population; specifically, the LD decay becomes slower and the SFS is shifted toward high-frequency derived variants for migration rates that are intermediate ($4Nm = 3$, where N is the effective population size and m the probability of migration per individual and per generation; figure 5 in [31]). Similar results are obtained from island models.

It is generally believed that, unlike the localized effect of a selective sweep, neutral demographic changes generate genome-wide patterns. This idea of “local sweep effects” *vs.* “global demographic effects” in the genome has been extensively used to control the demography-induced false positive rates [56, 65, 73]. In SFS-based sweep scans, this idea translates to a two-step computational approach that entails the initial estimation of an average, genome-wide SFS (background SFS) followed by a detection step, for those genomic regions that fit the selection model better than the background SFS. An issue with such an approach, however, is that it does not take into account the fact that SFS is characterized by great variation along the genome. In bottlenecks, or in models with gene flow, which generate great variance along a recombining chromosome [13, 26, 70, 93], the usage of the average, genome-wide SFS may be problematic. Therefore, under certain bottleneck demographic scenarios, there can be neutral-like genomic regions, as well as sweep-resembling ones, regardless of the actual existence of a selective sweep. Since both recombination and the alternation of genealogies along a recombining chromosome are stochastic, it is highly challenging to determine which genealogies are shaped by neutrality and which genealogies are shaped by positive selection. Current approaches are not able to completely overcome the confounding effect of bottlenecks on positive selection in recombining chromosomes, therefore users should be careful when interpreting results of selective sweep scans. It should be noted however, that several tools, such as SweepFinder, SweepFinder2, SweeD, OmegaPlus, and RAiSD and/or the deployment of the demographic model as the null model, contribute to alleviating the problem generated by the confounding effects of demography.

Demography not only affects the false positive rate (FPR) of the detection methods, but it also affects the true positive rate (TPR). This derives from the fact that the SNP patterns that emerge from the combined action of *demography and selection* are unknown. For instance, the SFS-based tools SweepFinder and SweeD (presented in a following section) assume that if a lineage escapes the selective sweep due to a recombination event, then, prior to the sweep, its frequency is given by the neutral (or background) SFS. This is valid if the selective sweep has occurred in a constant-size population. If, however, the population has experienced population size changes (or other demographic events such as migrations), this assumption does not necessarily hold.

Given the challenges that demographic changes pose to the accurate detection of positive selection, it is unfortunate (even though expected) that most natural populations have experienced various demographic scenarios during their evolutionary history. For example, the European population of *D. melanogaster* experienced a severe bottleneck about 15,800 years ago, when the European population diverged from the African population [56]. The duration of the bottleneck was about 340 years and the effective population size during the bottleneck was only 2200 individuals [56], thus the effective population size of the European population was decreased by a factor of 500, approximately. Regarding the demography of human populations, the proposed models suggest several bottleneck (founder) events and migrations between subpopulations [36]. Domesticated animals have also experienced a series of bottleneck events during the domestication process. Using only mtDNA and the approximate Bayesian computation methodology, Gerbault et al. [40] report that goats have experienced severe bottleneck events during their domestication. Approximate Bayesian computation was also used to provide insights into the demographic history of silkworm [99]. Using 17 loci in the domesticated silkworm, they reported that the most plausible scenario explaining the demographic history of silkworm comprises both bottleneck and gene flow events [99].

4 A Guideline on Selection Detection Tools

4.1 Summary Statistics

Summary statistics are computationally inexpensive data calculations. On whole-genome data, typically they are applied following a sliding window approach. Simpler statistics such as Tajima's D or the SNP count do not require phased data, but only SNP calling, whereas LD-based ones require phased data. Several summary statistics serve as neutrality tests because their distributions are affected by the presence of positive selection (for example, Tajima's D obtains negative values in the proximity of a strongly beneficial allele).

Relying on Tajima's D , Braverman et al. [22] were able to detect genomic regions affected by recent and strong positive selection in simulated datasets, as well as to demonstrate that in regions of low genetic diversity and low recombination rate (e.g., around centromeres or at telomeres) a simple hitchhiking model is not a sufficient explanation for the observed DNA polymorphisms. Since then, Tajima's D has been deployed in numerous studies as a neutrality test to detect selection [12, 19, 20, 58, 67, 79, 94]. This summary statistic captures the difference between two estimates of the diversity level $\theta = 4N_e\mu$, where μ is the mutation rate. The first estimate, π , is based on the number of pairwise differences between sequences, while the second one, Watterson's θ (θ_W), is based on the number of polymorphic sites. Tajima's D obtains negative values in the proximity of a selective sweep, since π decreases with both high- and low-frequency derived variants, while θ_W remains unaffected.

In 2000, Fay and Wu [37] proposed a new statistic, H , which obtains low values in regions where high-frequency derived variants are overrepresented. To distinguish between high- and low-frequency derived variants, Fay and Wu's H relies on an out-group sequence. Additionally, Fay and Wu [37] invented a new unbiased estimator for θ , named θ_H , which assumes high values in regions with overrepresented high-frequency derived variants. The H statistic is defined as the difference between π and θ_H , and as such it becomes significantly negative in the proximity of a beneficial mutation. Since a back-mutation will result in the incorrect inference of the derived polymorphic state, Fay and Wu's H requires the probability of mis-inference to be incorporated in the construction of the null distribution of the statistic. In 2006, Zeng et al. [101] improved the H statistic by adding the variance of the statistic in the denominator, thus scaling H by the variance of the statistic.

Depaulis and Veuille [34] introduced two neutrality tests relying on haplotypes. The first summary statistic, K , is simply the number of distinct haplotypes in the sample. In the presence of a selective sweep K takes low values. The second test measures haplotype diversity, denoted by H (or DVH, Depaulis and Veuille H , to be distinguished from Fay and Wu's H). DVH is calculated as $DVH = 1 - \sum_{i=1}^K p_i^2$, where p_i is the frequency of the i th haplotype. Both the DVH and the K summary statistics are conditioned on the number of polymorphic sites, s , which yields the construction of the null (neutral) distribution of the statistic rather problematic. Depaulis and Veuille simulated data using a fixed number of polymorphic sites s , and without conditioning on the coalescent trees. This approach is suboptimal because the number of polymorphic sites is a random variable that follows a Poisson distribution, and it is determined by the total length of the (local) coalescent tree and the mutation rate. Thus, to construct the null distribution of the statistic, a two-step approach is required: first, a coalescent tree is

generated according to the demographic model and mutations are placed randomly on its branches (this step can be achieved using Hudson's *ms* [45]), and second, a rejection process is applied in order to condition on the number of polymorphic sites s , during which only the simulations that produced s segregating sites are kept while the rest are discarded. Thus, only a subset of coalescent trees will be accepted: the trees that given the mutation rate result in the specified number of segregating sites s .

Typically, summary statistics are applied on whole-genome data following a sliding-window approach. This allows efficient computations on large datasets for those statistics used as neutrality tests, introducing, however, two main problems. The fixed size of the window length creates the first problem since small changes (even by only a few bases) of the window length may shift the results from statistically non-significant to significant [72], regardless of whether the window size is measured in number of base pairs or number of SNPs. The second problem, which is common for most neutrality tests, is that they are not robust to demographic changes of the population. For instance, Tajima's *D* can assume negative values in a population expansion scenario as well as locally in genomic regions under a bottleneck scenario. It also becomes negative in genomic regions that have experienced purifying selection and in regions affected by positive selection. Fay and Wu's *H* can become negative in demographic models that increase the high-frequency derived variants. Such demographic models include gene flow [31] or sampling from one deme that is part of a metapopulation [87].

4.2 Detecting Sweeps in Whole Genomes

The advent of next generation sequencing (NGS) allowed the analysis of whole genomes at different geographic locations and environmental conditions, and revealed a need for more efficient processing solutions in order to handle the increased computational and/or memory requirements generated by large-scale NGS data. While typical summary statistics are generally suitable for NGS data, they are applied on fixed-size windows, and as a result they do not provide any insight on the extent of a selective sweep. More advanced methods that rely on the CLR test (e.g., SweepFinder [65], SweepFinder2 [33], and SweeD [73]) or on patterns of LD (e.g., OmegaPlus [6, 7]) perform an optimization on the size of the window and, therefore, they provide information on the genomic region affected by a selective sweep at the cost of increased execution times. The aforementioned methods have been widely used to detect recent and strong positive selection in a variety of eukaryotic or prokaryotic organisms, such as human [18, 65, 75], *D. melanogaster* [11, 25, 95, 98], lizards [54], rice [24], butterflies [59], and bacteria [63].

4.2.1 *SweepFinder*

In 2005, Nielsen et al. [65] released SweepFinder, an advanced method to detect selective sweeps that relies on information directly derived from the SFS, either folded or unfolded. SweepFinder implements a composite likelihood ratio (CLR) test. The numerator of SweepFinder represents the likelihood of a sweep at a given location in the genome, given its selection intensity α . The denominator accounts for the neutral model. An important feature of SweepFinder is that neutrality is modeled based on the empirical SFS of the entire dataset. All SNPs are considered independent, therefore allowing the likelihood score per region for the sweep model to be computed as the product of per-SNP likelihood scores over all SNPs in a region. SweepFinder was among the first software releases with the capacity to analyze whole genomes via a complete and standalone implementation. SweepFinder can process small and moderate sample sizes efficiently. However, the source code does not handle floating-point exceptions that occur when a large number of sequences are analyzed, yielding analyses with more than 1027 sequences impossible.

4.2.2 *SweeD*

Pavlidis et al. [73] released SweeD (**S**weep **D**etector), a stable, parallel, and optimized implementation of the same CLR test as SweepFinder. SweeD can parse various input file formats (e.g., Hudson's ms, FASTA, and the Variant Call Format) and provides the option to employ a user-specified demographic model for the theoretical calculation of the expected neutral SFS. Also, it allows the user to provide her/his own points of interest where the CLR will be assessed (via the gridfile option). Pavlidis et al. [73] showed that sweep detection accuracy increases with an increasing sample size, and altered the mathematical operations for the CLR test implementation in SweeD to avoid numerical instability (floating-point underflows), allowing the analysis of datasets with thousands of sequences. The time-efficient analysis of large-scale datasets in SweeD is mainly due to two factors: (a) parallel processing using POSIX threads, and (b) temporary storage of frequently used values in lookup tables. Additionally, SweeD relies on a third-party library for checkpointing (Ansel et al. [10]) to allow resuming long-running analyses that have been abruptly interrupted by external factors, such as a power outage or a job queue timeout.

4.2.3 *SweepFinder2*

More recently, DeGiorgio et al. [33] released SweepFinder2. SweepFinder2 uses the statistical framework of SweepFinder, and additionally it takes into account local reductions in diversity caused by the action of negative selection. Therefore, it provides the opportunity to distinguish between background selection and the effect of selective sweeps. Thus, it exhibits increased sensitivity and robustness to background selection and mutation rate variations. Besides the ability to account for reductions in the diversity caused by background selection, the implementation of SweepFinder2 is

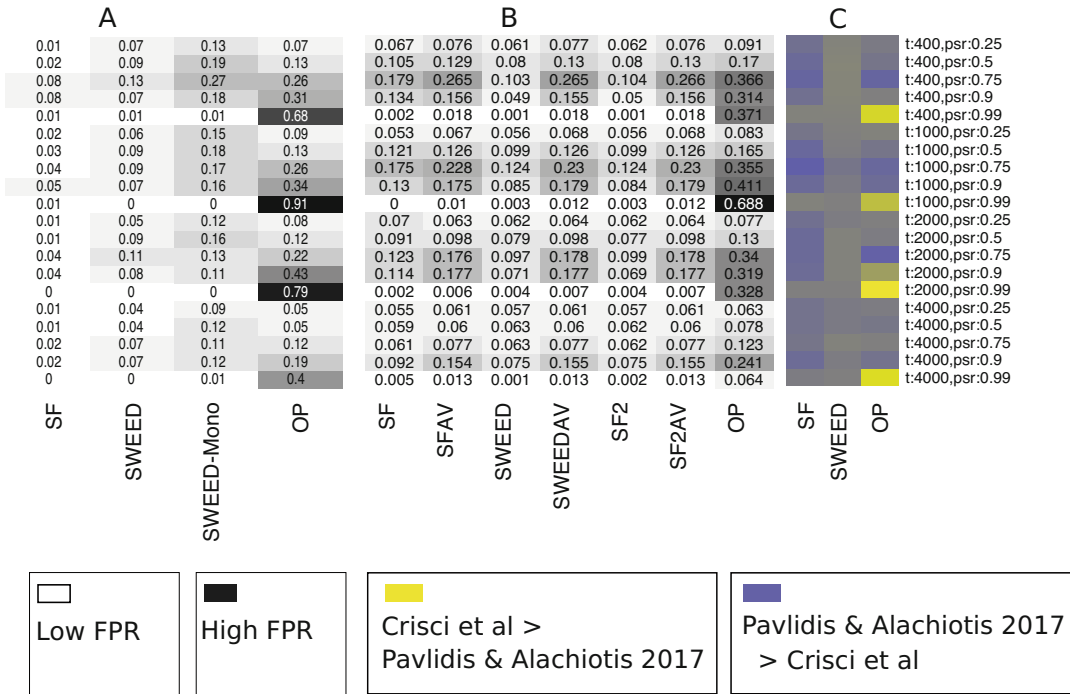


Fig. 3 False positive rates for the selective sweep detection process under various algorithms and demographic models. Demographic models consist of bottlenecks and are characterized by two parameters: t is the time in generations since the recovery of the populations, and psr the relative population size reduction during bottleneck. Prior to the bottleneck, the population size equals to the present-day population size. We show the results from the study of Crisci et al. [30] (a), our analysis in the current study (b) and the difference between a and b (c). Note that Crisci et al. studied SweepFinder (SF), SweeD (SWEED), SweeD with monomorphic (SWEED-Mono), and OmegaPlus (OP). In the current work, we studied SweepFinder (SF), SweepFinder with average SFS (SWEEDAV), SweeD (SWEED), SweeD with average SFS (SWEEDAV), SweepFinder2 (SF2), SweepFinder2 with average SFS (SF2AV), and OmegaPlus. Thus, in (c) we show only results from the common tools (SF, SWEED, OP). In (a) and (b), the darker a cell, the lower the false positive rate. In (c), yellow denotes that Crisci et al. report higher false positive rate than [69] while blue denotes that the reported false positive rate by Crisci et al. is lower. The figure is adapted from [69]

very similar to SweepFinder. However, there exist code modifications that increase the stability of SweepFinder2 on the calculation of likelihood values. Using simulated data with constant mutation rate and in the absence of negative selection, SweepFinder2 scores are closer to those obtained by SweeD rather than the initial SweepFinder implementation (see Fig. 3).

4.2.4 OmegaPlus

In 2012, Alachiotis et al. [7] released a high-performance implementation of the ω -statistic [51] for the detection of selective sweeps by searching for a specific pattern of LD that emerges in the neighborhood of a recently fixed beneficial mutation. The ω -statistic assumes a high value at a specific location in the genome, which can be indicative of a potential selective sweep in the region,

if extended contiguous genomic regions of high LD are detected on both sides of the location under evaluation, while the level of LD between the high LD regions remains relatively low. OmegaPlus evaluates multiple locations along a dataset following an exhaustive per-region evaluation algorithm, which was initially introduced by Pavlidis et al. [71]. The algorithm by Pavlidis et al. [71] required large memory space for the analysis of many-SNP regions and exhibited increased complexity, yielding the analysis of regions with thousands of SNPs computationally unfeasible. OmegaPlus introduced a dynamic programming algorithm to reduce the computational and memory requirements of the exhaustive evaluation algorithm, enabling the efficient analysis of whole-genome datasets with millions of SNPs. OmegaPlus exhibits a series of four different parallelization alternatives [4, 6] for the distribution of computations to multiple cores to overcome the load balancing problem in selective sweep detection due to the difference in SNP density between regions in genomes.

4.2.5 MFDM Test

In 2011, Li et al. [55] presented a neutrality test that detects selective sweep regions using the maximum frequency of derived mutations (MFDM), which is a paramount signature of a selective sweep. According to [55], the MFDM test is robust to processes that occur in a single and isolated population. This is because there is no demographic scenario in single and isolated populations that generates a non-monotonic SFS and increases the amount of high-frequency derived variants. Thus, at least in theory, the test is robust to demographic models, such as bottlenecks, when they occur in isolated populations. However, four severe problems arise regarding the robustness of the test, which broadly apply to other tests of neutrality as well: (1) although bottlenecks generate monotonic *average* SFSs, certain genomic regions may locally exhibit increased amounts of high-frequency derived variants, even in the absence of positive selection, (2) high-frequency derived variants are a signature of selective sweeps in constant populations but it is not known whether and how they will be affected by the combined action of selection and demography, (3) in populations that exchange migrants with other demes (non-isolated), the frequency of high-frequency derived variants may increase (e.g., [31]), and (4) back-mutations (in general, the violation of the infinite-site model) may also increase the amount of high-frequency derived variants.

4.3 RAiSD

In 2018, Alachiotis and Pavlidis [5] introduced the μ statistic and released RAiSD (Raised Accuracy in Sweep Detection). The μ statistic is a composite evaluation test that scores genomic locations by relying on the enumeration of SNP vector patterns (entire alignment columns) to quantify changes in the SFS, the levels of LD, and the amount of genetic variation. RAiSD implements a SNP-driven, sliding-window algorithm that reuses calculated data

between overlapping windows to considerably reduce execution times. It exhibits increased detection accuracy and sensitivity due to the fact that consecutive SNP windows with variable size in terms of base pairs are placed along a dataset with a step of 1 SNP. This achieves increased granularity in SNP-dense regions and avoids redundant operations in SNP-sparse ones, consequently improving processing speed without deteriorating the quality of the results. Furthermore, RAI_{SD} couples the sliding window algorithm with an out-of-core approach that allocates a negligible amount of memory (typically few MBs) irrespectively of the dataset size, thus maintaining overall low memory requirements. Details on RAI_{SD} software, its command line options, as well as working examples are available on the github repository of RAI_{SD} (<https://github.com/alachins/raisd>).

5 Evaluation

The aforementioned software tools (SweepFinder, SweepFinder2, SweeD, and OmegaPlus, and RAI_{SD} *see* Table 1) have been independently evaluated by three studies: Crisci et al. [30] studied the effect of demographic model misspecification on selective sweep detection, while Alachiotis and Pavlidis [4] conducted a performance comparison in terms of execution time for various dataset sizes and number of processing cores. Alachiotis and Pavlidis [5] evaluated all tools in terms of detection accuracy, sensitivity, and execution time, with the aim to assess RAI_{SD}. We summarize these results in the following subsections and partially reproduce the FPR evaluation analysis by Crisci et al. [30], including SweepFinder2.

Table 1
List of software tools for selective sweep detection

	Method	Implementation	Availability (source code , web service)
SweepFinder (2005)	SFS	Sequential	http://people.binf.ku.dk/rasmus/webpage/sf.html , –
OmegaPlus (2012)	LD	Parallel	https://github.com/alachins/omegaplus , http://pop-gen.eu
SweeD (2013)	SFS	Parallel	https://github.com/alachins/sweed , http://pop-gen.eu
SweepFinder2 (2016)	SFS	Sequential	http://www.personal.psu.edu/mxd60/sf2.html , –
RAI _{SD} (2018)	Mixed	Sequential	https://github.com/alachins/raisd , –

The table is adapted from [69]

5.1 Detection Accuracy

Crisci et al. [30] calculate the FPR for the neutrality tests using the following pipeline: (1) simulations from equilibrium models using Hudson's ms [45] and constant number of SNPs. This set of simulations is used only for the determination of the thresholds for the tools; (2) simulations using sfscode [44] (constant or bottlenecked population). These data are called empirical datasets, and are used for the estimation of the FPR; (3) execution of the neutrality tests on the empirical datasets. The FPR is estimated by assigning each empirical dataset to a threshold value from an equilibrium model with similar number of SNPs. Note that, such an approach differs from the approach that has been followed by other studies (e.g., [38, 60]), where the null model is specified by the inferred neutral demographic model. Specifying the null model by the inferred neutral demographic model controls efficiently for the FPR. Thus, Crisci et al. effectively studied how demographic model misspecification affects the FPR. Another major difference between the approach followed by Crisci et al. and other studies is that, for the SFS-based methods (SweepFinder, SweeD), Crisci et al. calculate the neutral (or *prior-to-sweep*) SFS using the candidate region itself (here 50 kb), instead of the average SFS on a chromosome-wide scale. Even though the first approach might have a lower FPR, the later is more powerful to detect selective sweeps: when the neutral SFS is calculated by a small genetic region that potentially includes a sweep, the affected (by the sweep) SFS is assumed to represent neutrality. Thus, the CLR test will assume lower values. For neutral equilibrium models, i.e., constant population size, they find that the FPR for SweepFinder ranges from 0.01 to 0.18, depending on the mutation and recombination rate: the lower the mutation and recombination rates, the higher the FPR of SweepFinder. The FPR for SweeD ranges between 0.04 and 0.07. For OmegaPlus, the FPR ranges between 0.05 and 0.07. In general, the FPR for all tools is low when the demographic model is at equilibrium.

When the assumption of an equilibrium population is violated and the empirical datasets are derived from bottlenecked populations, the FPR increases. Such an increase of the FPR is more striking when the average SFS of the empirical dataset is used to represent the SFS of the null model. The reason for such an increase is that bottlenecked datasets show great variance of the SFS from a region to another. Thus, even though, on average, a bottlenecked population will have a monotonically decreasing SFS [104], there might be regions that show an excess of high-frequency and low-frequency derived variants, and thus they mimic the SFS of a selective sweep.

Interestingly, Crisci et al. report low FPR for SweepFinder and SweeD. For OmegaPlus, they report high FPR for the very severe bottleneck scenario, where the population size has been reduced by 99%. For SweepFinder and SweeD, the FPR ranges between 0 and 0.08, and 0 and 0.13, respectively. For OmegaPlus, they report

FPR between 0.05 and 0.91. We repeated the analysis of Crisci et al. for SweeD, SweepFinder, and OmegaPlus, including SweepFinder2. Furthermore, we have included execution results of SweepFinder, SweeD, and SweepFinder2 using the average SFS instead of the regional SFS. We used Hudson’s *ms* for *all* simulations, whereas Crisci et al. had used *sfs_code* for the empirical simulated data. In general our results are comparable to Crisci et al., but we report higher FPR than Crisci et al. A notable exception is the case of OmegaPlus in the severe bottleneck case, where our FPR is considerably lower. Perhaps this is due to the simulation software, as we used Hudson’s *ms* (coalescent) simulator, while Crisci et al. used *sfs_code* (forward). FPR results are shown in Fig. 3.

Since FPR is considerably increasing when a false model (e.g., equilibrium) is used to construct the null hypothesis, we repeated the aforementioned analysis using a bottleneck demographic model. Using a bottleneck demographic model for the construction of the null hypothesis reduces the FPR to very low values (Fig. 4). Here, we have used the bottleneck model characterized

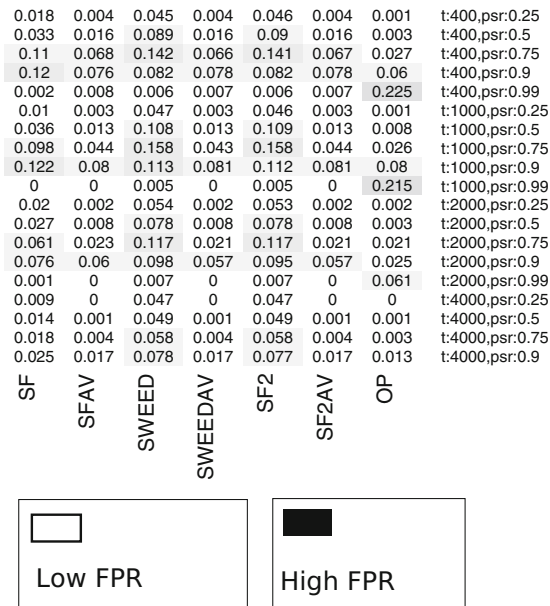


Fig. 4 False positive rates for the selective sweep detection process under various algorithms and demographic models when the demographic model used for the construction of the threshold value is a bottleneck model instead of an equilibrium model. *t*: time since the population size recovery (generations). *psr*: relative population size reduction during bottleneck. To compute *all* threshold values, we have used the bottleneck model characterized by a population recovery at time $t=1000$ generations, and bottleneck population size reduction by 0.90. The duration of the bottleneck was 4000 generations. FPR values have been reduced considerably compared to the case that the equilibrium model was used for the calculation of the threshold values (Fig. 3). The figure is adapted from [69]

by a population size reduction of 0.99, a recovery time of 1000 generations, and bottleneck duration of 4000 generations, even though empirical datasets were composed by additional models. The ancestral population size was equal to the present-day population size.

Regarding the true positive rate (TPR), Crisci et al. report that under strong selection in an equilibrium population ($2N_e s = 1000$, where s is the selection coefficient), TPR for SweepFinder and SweeD is moderate and ranges between 0.32 and 0.34. For OmegaPlus, TPR is higher and equals to 0.46. For weaker selection ($2N_e s = 100$), OmegaPlus is still the most powerful tool to detect selective sweeps. For selective sweep models in bottlenecked populations, OmegaPlus outperforms SFS-based methods and it is the only test studied by Crisci et al. able to detect selective sweeps. Finally, regarding recurrent hitchhiking event (RHH), OmegaPlus reports higher values of TPR.

5.2 Execution Time

The performance comparisons conducted by Alachiotis and Pavlidis [4] aimed at evaluating the effect of the number of sequences and SNPs on execution time, as well as the capacity of each code to employ multiple cores effectively to achieve faster execution. Table 2 shows execution times on a single processing core for different dataset sizes, ranging from 100 sequences to 1000 sequences, and from 10,000 SNPs up to 100,000 SNPs. Additionally, the table provides (in parentheses) how many times faster are SweeD and OmegaPlus than SweepFinder.

The comparison between SweepFinder and SweeD is the most meaningful one since both tools implement the same floating-point-intensive CLR test based on the SFS, thus requiring the same type and amount of arithmetic operations. The significantly faster execution of OmegaPlus on the other hand, which relies on LD, is attributed to the fact that a limited number of computationally intensive floating-point operations are required, with the majority of operations being performed on integers, such as the enumeration of ancestral and derived alleles.

The execution times in Table 2 refer to sequential execution. Multiple cores can be employed by SweeD and OmegaPlus,

Table 2
Comparison of execution times (in seconds) for different dataset sizes (format: D—number of sequences—number of SNPs) on a single processing core [4]

	D— 10^2 — 10^4	D— 10^2 — 10^5	D— 10^3 — 10^4	D— 10^3 — 10^4
SweepFinder	540 (1×)	4138 (1×)	132,938 (1×)	135,996 (1×)
SweeD	125 (4.3×)	1169 (3.5×)	283 (469×)	1345 (101×)
OmegaPlus	6 (90×)	652 (6.4×)	7 (18,991×)	753 (180×)

achieving speedups that vary depending on the number of sequences and SNPs. The parallel efficiency of SweeD decreases with an increasing sample size, whereas the respective parallel efficiency of OmegaPlus increases. As the number of SNPs increases, both SweeD and OmegaPlus exhibit poorer parallel efficiency, which is attributed to load balancing issues that arise with an increasing variance in the SNP density along the datasets.

6 Machine Learning for Population Genetics

6.1 *Machine Learning Background*

One of the main problems of model-based methods, such as SweeD [73], SweepFinder [65], and OmegaPlus [7], is their inability to provide accurate results when their assumptions are violated. Since, however, in natural populations several of the assumptions of model-based methods (e.g., constant population size) are violated, there is a need for more flexible methodologies. Machine learning was introduced in population genetics as an alternative methodology to detect genomic regions that evolve under selection by treating the problem of detecting selection as a classification problem [83].

The inspiration behind the field of machine learning (ML) was the concept of artificial intelligence (AI). In AI, the main goal was to successfully recognize patterns previously unseen by the algorithm. For this purpose, the process of learning began via observing examples to search for patterns in data and attempt to improve decisions in the future based on the provided examples. The aim is for computers to learn, or rather be trained, by these examples without human assistance, similarly to how humans, and many other living organisms learn from experience. ML enables the analysis of massive quantities of data. The data used in ML tasks can be split into two categories: training data and test data. Training data are used for learning, whereas test data are used to test/evaluate performance, or, in other words, how well the algorithm learned to work for the given task.

6.2 *Categories of Machine Learning*

The field of ML can be split into three different categories in terms of the learning approach. The first category is supervised learning, which is concerned with predicting the value of a response variable or label (either a categorical or continuous value) on the basis of the input variables/features. Supervised learning accomplishes this feat through the use of a training set of labeled data examples whose true response values are known. The second category is unsupervised machine learning, where, contrary to supervised learning, these learning algorithms are used when the information in the training set is neither classified nor labeled. Unsupervised learning studies how systems can infer a function to describe a hidden structure from unlabeled data. The system does not infer the classes

of the data, but it explores the data and infers hidden structures. The third category is reinforcement learning, a field strongly linked to artificial intelligence and game theory. Reinforcement is a learning method that interacts with its environment by producing actions and discovering errors or rewards. Trial-and-error search and delayed reward are the most relevant characteristics of reinforcement learning. This method allows machines and software agents to automatically determine the ideal behavior within a specific context to maximize its performance. Simple reward feedback is required for the agent to learn which action is the best. A further categorization can be made between classification and regression tasks. Classification deals with identifying a group membership where the output variable takes class labels. Regression involves predicting a response when the output variable takes continuous values.

For an in-depth description of machine learning, Alpaydin's introduction to machine learning [66], Michel's machine learning [61], and Bishop's pattern recognition and machine learning [64] are highly recommended.

6.3 Algorithms in Machine Learning

There are various approaches to train machines, ranging from basic decision trees to multilayer artificial neural networks (which evolved to deep learning), depending on what task should be accomplished and the type and amount of available data. Here, we investigate the performance of various well-known and widely used ML algorithms in the classification problem of selection versus neutrality. Our goal is to examine whether machine learning algorithms, used in population genetics analyses, can accurately infer selection. Classification algorithms can be either generative or discriminative. A generative algorithm models how the data was generated in order to categorize them in different classes. Thus, its aim is to find the category that is most likely to generate the observed result. A discriminative algorithm does not care about how the data was generated, it simply categorizes the given set of features. A general concern of the ML-related problems is overfitting. Overfitting [43] is the phenomenon when results of training cannot reliably capture previously unseen data due to being tailored on just the given training data. In other words, if a model performs significantly better on the training data than on unseen/test data, then the model probably suffers from overfitting.

In this study, the ML classifiers that we will evaluate are logistic regression (LR), random forests (RF), k-nearest neighbors (kNN), and support vector machines (SVM). The ML framework was implemented in Python using the *sklearn* [74] package.

Naive Bayes A generative classifier that uses the Bayes rule to describe the joint probability of data and classes is called naive Bayes (NB). An important drawback of NB is that it assumes conditional independence of features given the class label. However, in population genetics, conditional independence does not hold for most of the features neither under neutrality nor under selection. Thus, naive Bayes may not be appropriate for population genetics data, and we do not evaluate it in this study.

Logistic Regression It is a classifier that assumes a parametric form for the distribution $P(Y|X)$ and directly estimates its parameters from the training data. The central premise of LR is the assumption that the input space can be separated into two regions, one for each class, by a linear boundary. Unlike NB, LR does not assume that the features are conditionally independent.

k-Nearest Neighbor kNN is not strictly a learning classifier but rather a memory-based classifier. It classifies each of the test data by its position based on its k closest/nearest neighbors for which the class is known. To the best of our knowledge, kNN has not been examined as a selection/neutrality classification algorithm yet.

Random Forests RF is a classifier that works well for classification problems as it is able to exploit both high- and low-“informative” features and to deal with the problem of overfitting. The original classification algorithm that inspired RF was the decision trees method. Based on the values each of the features may take, “decision” nodes are created resulting in a tree structure. Upon reaching a leaf of this tree, a decision is achieved for the label of the input data. The features with lower entropy (most informative) appear closer to the root of the tree. However, a single tree might be heavily biased and as a result the algorithm may overfit. The solution to the overfitting problem is RF, a classifier that consists of several different decision trees whose outcomes are combined, usually by averaging the results, to predict the class of the input.

Support Vector Machines It is a machine learning algorithm proposed by Cortes et al. [28]. SVMs attempt to split the dataset into two classes via using a hyperplane that separates those classes. The goal is to find the ideal hyperplane that best separates those classes. It uses specific data points of each class to determine the position of the hyperplane. These points are called the support vectors. A distance between the hyperplane and the closest support vector from each class is kept, namely the margin. SVMs attempt to maximize this margin to maximize the probability of correctly classifying new data. Due to the ability of SVMs to reach higher dimensions, they do not suffer from the “curse of dimensionality,” making them a suitable algorithm for classifying between selection and neutrality.

7 Methods

7.1 Data Generation

There exist various models that produce single nucleotide polymorphisms (SNP) from demographic models. To generate our data, we used the *ms* tool, a Monte Carlo computer program written in C, that generates samples drawn from a population evolving according to a Wright–Fisher neutral model [42]. The program assumes an infinite-sites model of mutation, and allows recombination, gene conversion, symmetric migration among sub-populations, and a variety of demographic histories. For each sample, the program generates a random genealogical history of a segment of a chromosome. Conditional on the genealogy of a sample, mutations are randomly placed on the genealogy according to the usual assumption that the number of mutations on a branch is Poisson distributed with mean given by the product of the mutation rate and the branch length. The times between nodes in the genealogy are approximated by continuous (exponential) distributions.

We simulated neutral datasets and datasets with selection for 60 demographic models that include a variety of bottleneck scenarios (from mild to severe). For the selection data, we used an extension of *ms*, called *mssel*, kindly provided by R.R. Hudson. Each bottleneck model is characterized by a reduction in population size at some point in time and a recovery to the original population size (backwards in time). For each demographic model we generated 1000 datasets to incorporate the genealogical uncertainty in the training process. The mutation parameter of the model was set to $4N\mu = 2000$. In our simulations, we used a constant value for $4N\mu$. We could also sample this parameter from a distribution (e.g., Gaussian). Even though, results of the neutrality tests could be affected, at least partially, we expect that this effect will be minor because there is no direct involvement of the number of SNPs in the tests' results.

7.2 Computing Summary Statistics

The raw data, generated from *ms*, cannot be used directly for the classification task. Thus, from each polymorphic dataset, we compute a vector of summary statistics that will serve as data features. We used the software *CoMuStats* [68] to calculate a multitude of summary statistics from the *ms* simulations, such as Tajima's D [92], Wall's B and Q statistics [96], FST values [46], the site frequency spectrum [42], and others (Table 3).

Table 3
Description of a subset of the summary statistics generated by *CoMuStats*

Summary statistic	Definition
θ_W	Watterson’s estimator of θ using the number of segregating sites and the sample size
Tajima’s D	Computed as the difference between two measures of genetic diversity: the mean number of pairwise differences and the number of segregating sites, each scaled so that they are expected to be the same in a neutrally evolving population of constant size
B and Q	The number of pairs of adjacent segregating sites that are congruent
F_{ST}	A measure of population differentiation due to genetic structure. It is frequently estimated from genetic polymorphism data, such as single-nucleotide polymorphisms (SNP) or microsatellites
SFS	The number of segregating sites where the derived allele occurs i times out of n samples

7.3 Application of Classification Algorithms

7.4 Dataset Manipulation

When we obtain a collection of datasets, each belonging to an a priori known class, we follow the next steps for optimal and unbiased results. First, (1) we split the data in two parts. A part of the data is used for training, whereas the remaining is used as the test set. We used 20% of the generated dataset as a test set leaving 80% for training each model. Each classifier has parameters that need to be set before training begins. Thus, an important step of classification is to find the optimal parameter values. This process is called tuning. Thus, in step (2) we tune the classifier. Finally, in step (3) we evaluate the performance of the tuned classifier based solely on the (unseen) test set.

The simplest form of tuning is to use a part of the data for training and the remaining part for test. Tuning the parameters takes place by repeatedly evaluating the performance of the algorithm for different parameter values on the test set. This process, however, leads to overfitting. Another part of the dataset, which is named as validation set, is held in order to tackle the problem of overfitting. Using this approach, training proceeds only on the training set, while tuning the parameters of the classifier is performed on the validation set. When tuning is complete, a final evaluation can be done on the test set. This method is called cross validation (CV), and it remedies overfitting by ensuring that the parameters estimation of the classifier is not strictly associated with the data we used to estimate them. However, this simple approach results in tuning the classifier parameters based on a small part of the data, thus results may be suboptimal. A better strategy is the

so-called k -fold CV, in which the training set is split into k folds. We use all but one of the folds for training and the resulting model is validated on the remaining part of the data. This is repeated k times with a different validation set. The parameters of the classifier that result in the greatest accuracy, on average, are stored. As a final step, we train the classifier using the optimal parameters from the tuning step in the whole training set (training and validation). The accuracy is measured solely using the test set.

By using a single test set, the evaluation of the classification performance may be biased depending on the specific test set. Thus, an approach called nested k -fold CV can be followed. Nested k -fold CV effectively uses a series of train/validation/test set splits. In the inner loop (k -fold CV), the accuracy is approximately maximized by fitting a model to each training set, and then inferring the optimal parameter values using the validation set. In the outer loop (nested), the generalization error is estimated by averaging test set scores over several dataset splits.

Another popular method is the stratified nested k -fold cross validation, which ensures that representation of classes in each fold is according to their frequency in the original dataset. However, since our data are simulated, both classes are balanced (equally represented) by design and, therefore, there was no need to use stratified CV [77].

7.5 Feature Selection

To further increase the performance of our classifiers, we can use for training only those features (variables) that mostly enable classification between the two classes. In other words, by removing those features that do not contribute enough to the classification, the performance of the classifier will be increased. This method, which is widely used in machine learning, is called feature selection [39].

There are two problems related to feature selection. The first is how much does each feature contribute to solving the classification problem. Here, we use the mutual information [29] of each feature with respect to the others. The second problem is related to the number of features that will be used. This is performed via the SelectKBest package from python's sklearn [74]. In detail, we rank our features from the most informative one to the least informative one. We use the top m features ($2 \leq m \leq 40$) successively, and evaluate their performance.

8 Results

8.1 Reducing the Feature Space

We first perform the feature selection step. The number of features we kept was decided solely on the SVM classifier as described in Subheading 7.5. Each pair of datasets, one for selection and its neutral counterpart, was studied separately. We calculated the

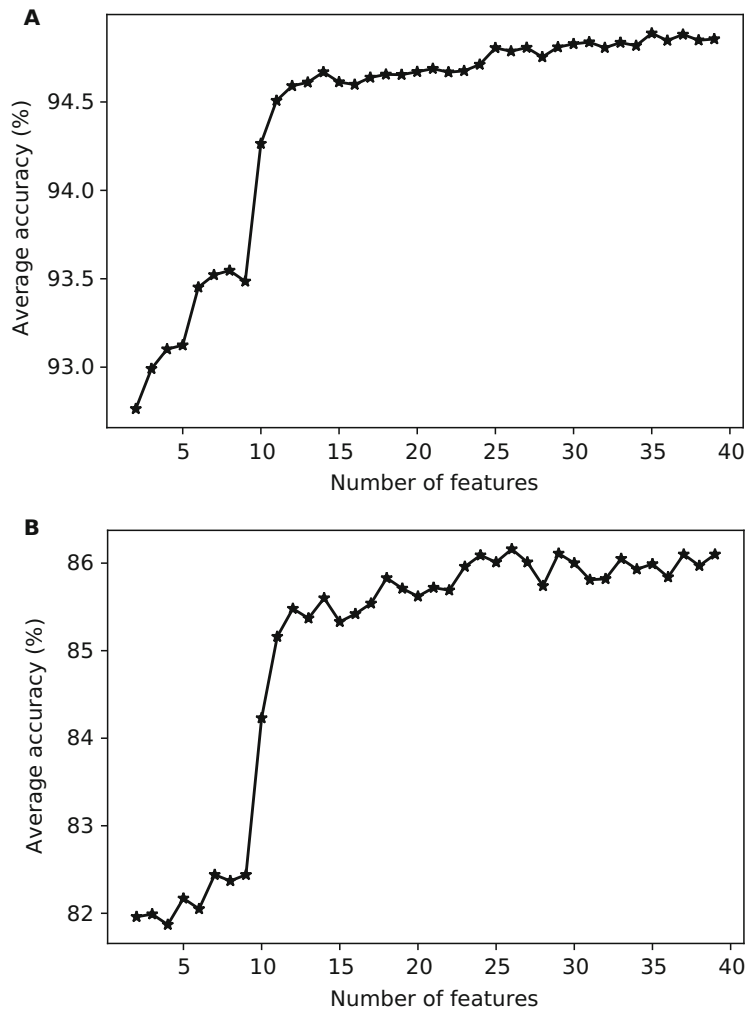


Fig. 5 Average accuracy of the feature selection procedure among the 60 datasets used in the study. (a) Average across all 60 datasets. (b) Average across the five datasets with the most severe bottleneck

average accuracy for each number of features across all 60 pairs. The results showed that 36 features produced the optimal results (94.865% average accuracy), as seen in Fig. 5.

The 60 datasets implement bottleneck demographic scenarios of various severities. Among them, some scenarios are mild, whereas others are severe. In mild bottlenecks, selection detection is a rather simple task. However, in severe bottlenecks disentangling selection from neutrality is challenging and often the accuracy of the algorithms is diminished [71]. To test the performance of the feature selection process in challenging scenarios, we chose to evaluate feature selection only on the five most severe bottleneck models. As seen in Fig. 5, the number of best performing features

was reduced to 26, achieving an average accuracy of 86.16%. Average accuracy is, as expected, lower overall since the best performing pairs were excluded.

8.2 Evaluation

Since the dataset is created from simulated data, we chose to have balanced classes by generating the same number of samples for both selection and neutrality. Thus, the trivial (random) classifier would achieve an average accuracy of 50%. All the classifiers were tested using the 36 best features.

8.2.1 Logistic Regression

Logistic regression works by separating the two classes using a linear boundary. It starts by setting the line according to the features. Then LR modifies the initial guess by changes its position or its slope to try and improve the accuracy of the classifier. A parameter to tune is the maximum number of attempts to optimize the accuracy. We set the parameter of maximum number of attempts to the values 100, 150, 200, and 250. To prevent the model from overfitting or underfitting, logistic regression uses a regularization penalty. The goal of that penalty is to not allow extreme values to influence the classifier. Two options for regularization are Ridge (L2 regularization) [21] and Lasso (L1 regularization) [21]. Ridge adds penalty equivalent to square of the magnitude of coefficients. Lasso adds a penalty equivalent to the absolute value of the magnitude of the coefficients. Both were considered during tuning, each with its own hyperparameters. Ridge uses sag [48] and lbfgs [9]. Lasso uses saga [32] and liblinear [74].

The highest accuracy (94.92%) was achieved by using Ridge regression with saga for at most 150 attempts/iterations of the algorithm attempting to converge, whereas the performance dropped for more than 150 iterations (Fig. 6).

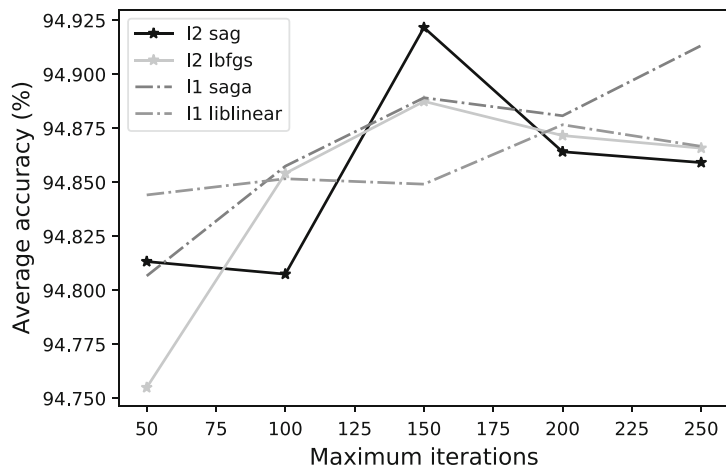


Fig. 6 Accuracy of logistic regression classifier for Lasso(L1) and Ridge (L2) regularization, while increasing maximum iteration allowed in order to converge

8.2.2 Random Forests

For the random forests classifier, the tuning parameters are the maximum depth the tree was allowed to reach, the maximum number of features to consider for each split, and the number of decision trees generated. Forests consisting of 50, 100, 150, and 200 trees were examined. For these trees, a maximum depth of 10, 20, 30, and 36 splits was allowed. For each split, either the square root (F_{SQRT}) or the logarithm (F_{LOG}) of the 36 features was the maximum number considered. We also used bagging, a method designed to improve the stability and accuracy of machine learning algorithms. According to [23], bagging is defined as:

Given a standard training set \mathbf{D} of size n , bagging generates m new training sets $\mathbf{D}\{\mathbf{i}\}$ each of size n' , by sampling from \mathbf{D} uniformly and with replacement. By sampling with replacement, some observations may be repeated in each $\mathbf{D}\{\mathbf{i}\}$. If $n' = n$, then for large n the set $\mathbf{D}\{\mathbf{i}\}$ is expected to have the fraction $(1 - 1/e)$ ($\approx 63.2\%$) of the unique examples of \mathbf{D} , the rest being duplicates. This kind of sample is known as a bootstrap sample. The m models are fitted using the above m bootstrap samples and combined by averaging the output (for regression) or voting (for classification).

As Fig. 7a, b shows, increasing the maximum depth of the decision trees, RFs achieve better accuracy up to a depth of 30 features. Further increasing the number of features results in a lower accuracy. Also, setting the maximum features considered to F_{SQRT} in each split performed better than F_{LOG} . A forest consisting of 150 trees performed optimally for both F_{SQRT} and F_{LOG} , and by comparing the two we can deduce that F_{SQRT} is the better performing method, as seen in Fig. 8.

8.2.3 K Nearest Neighbors

The two parameters to be tuned for the kNN classifier are the number of neighbors to consider and the distance metric used to calculate the distance between two neighbors. The two distance metrics under consideration are the Euclidean and the Chebyshev distance.

Euclidean is a better distance metric than Chebyshev for all neighbors considered (Fig. 9). By increasing the number of neighbors, the accuracy was increasing, reaching a maximum performance of 94.25% for 36 neighbors. For more than 36 neighbors, the accuracy declines.

8.2.4 Support Vector Machines

Support vector machines map the data to a predetermined high-dimensional space via a kernel function that enables classification of non-linearly separable data. The kernel function is used as a measure of similarity [81]. In particular, the kernel function $k(x, \cdot)$ defines the distribution of similarities of points around a given point x . $k(x, y)$ denotes the similarity of point x with another given point y . The polynomial kernel [81] and the random Bayesian forests (rbf) [81] are the kernels considered here. For the polynomial kernel, the maximum degree/dimension of the kernel

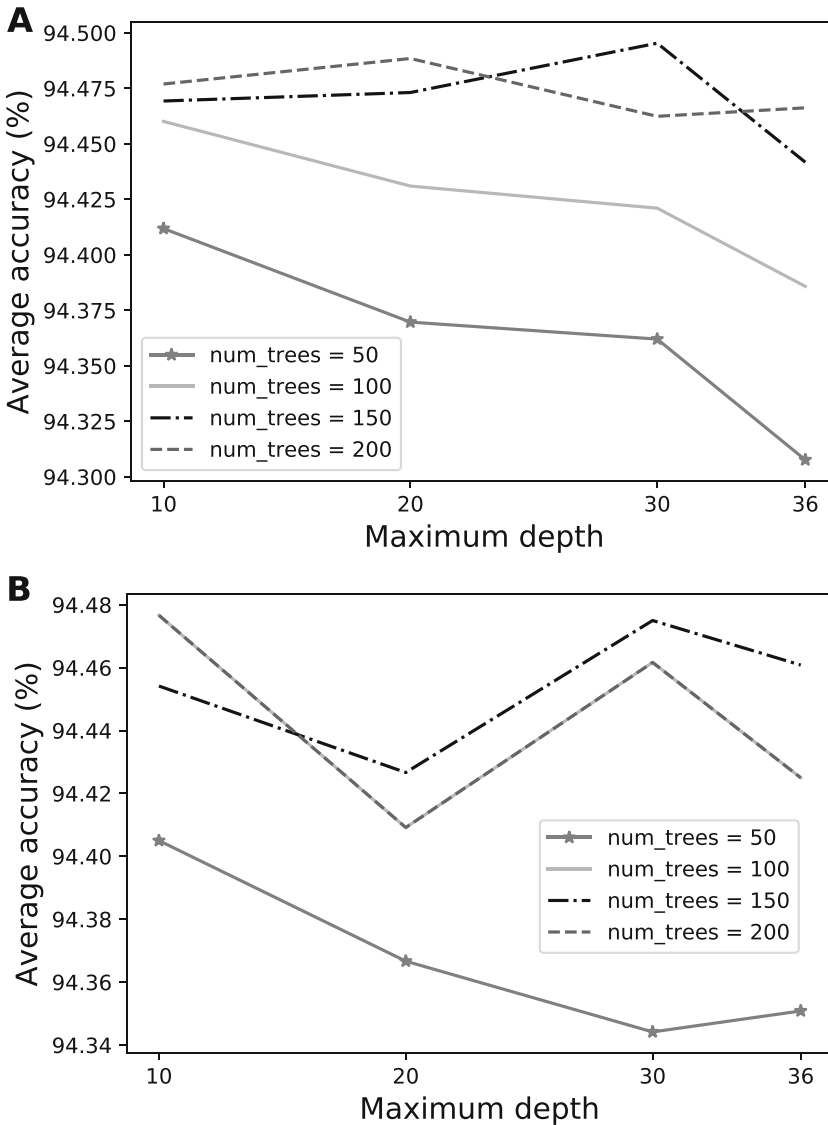


Fig. 7 Accuracy of random forest classifier for half, 75% and unlimited max depth allowed. Each line represents a different number of trees spawned (num_trees). **(a)** Log2 maximum features considered for each split. **(b)** Square root of maximum features considered for each split

function assumes the values 1 (the equivalent of a linear kernel), 2, 3, 4, and 5. For the rbf, the gamma hyperparameter ranged from -8 to 4 . In our setup, we use the Soft-Margin SVMs. Soft-Margin SVMs permit some errors while trying to find the optimal classification surface, thus the model is more robust to overfitting. Soft-Margin SVMs require a cost parameter that determines the number of errors we allow. The cost parameter ranges from 1 to 10 and its optimal value is 1. Based on Fig. 10, we can deduce that polynomial

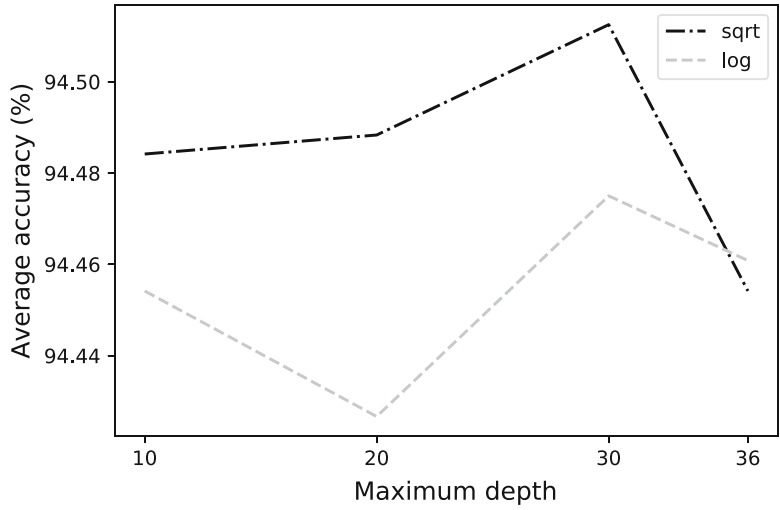


Fig. 8 Comparison of best performing cases (150 trees in the random forest) for log and sqrt maximum features

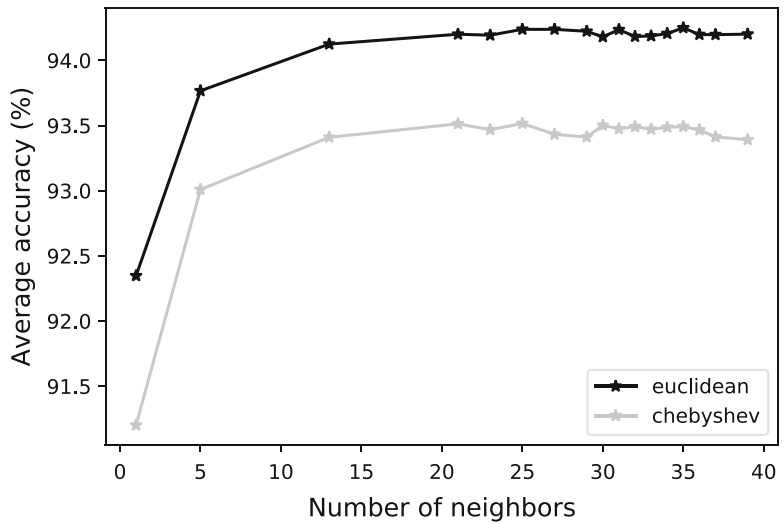


Fig. 9 Accuracy of kNN classifier for Euclidean and Chebyshev distances, for 1, 5, 21, 25, 27, 29, 31, 35, 36 neighbors

is the best performing kernel. It reaches the highest accuracy, 0.9484%, with a degree equal to 1.

The classifier that achieved the best overall performance is the support vector machines peaking at 0.9484%. Figure 11 compares the tuned versions of each classifier on each pair of datasets.

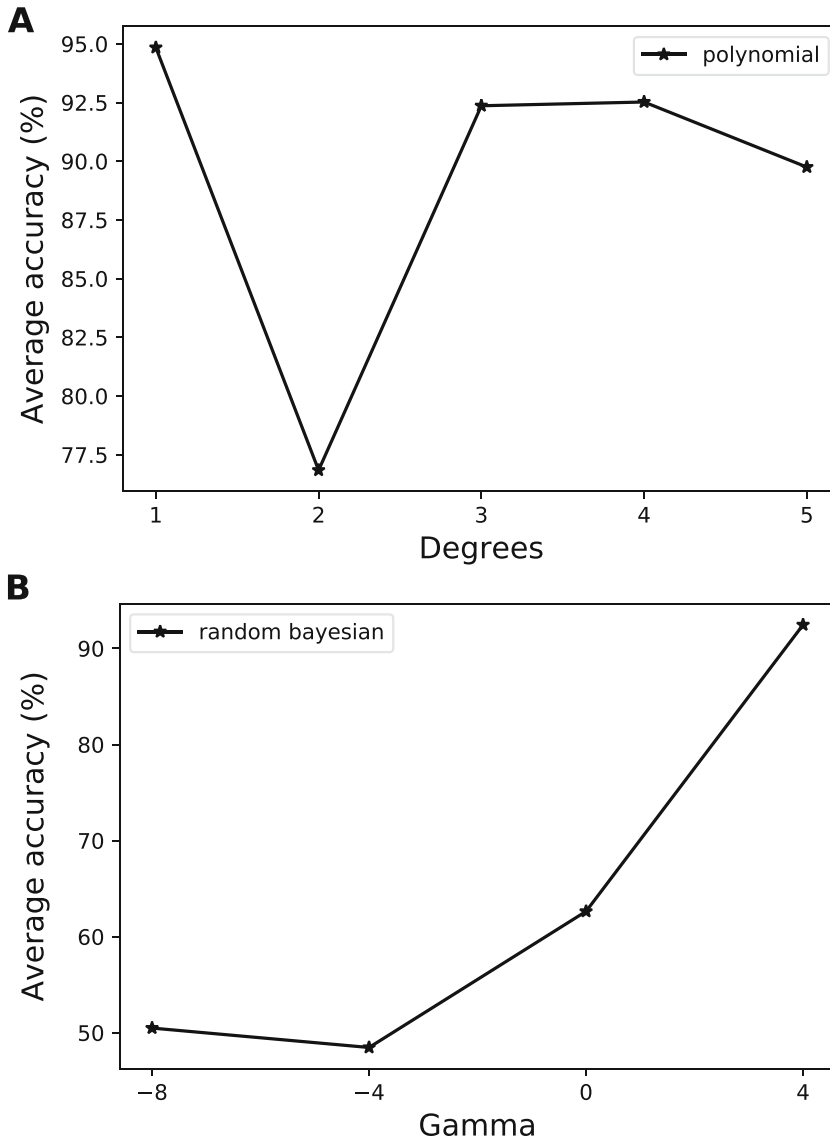


Fig. 10 Accuracy of support vector machines classifier for polynomial (a) and random Bayesian (b) forests kernel

9 Discussion

We demonstrate the use of machine learning algorithms in disentangling selection from neutrality. This task is treated as a supervised learning classification. We evaluated logistic regression, k -nearest neighbors, random forests, and support vector machines. All classifiers outperformed the trivial classifier and showed high accuracy, which, however, depends on the bottleneck severity.

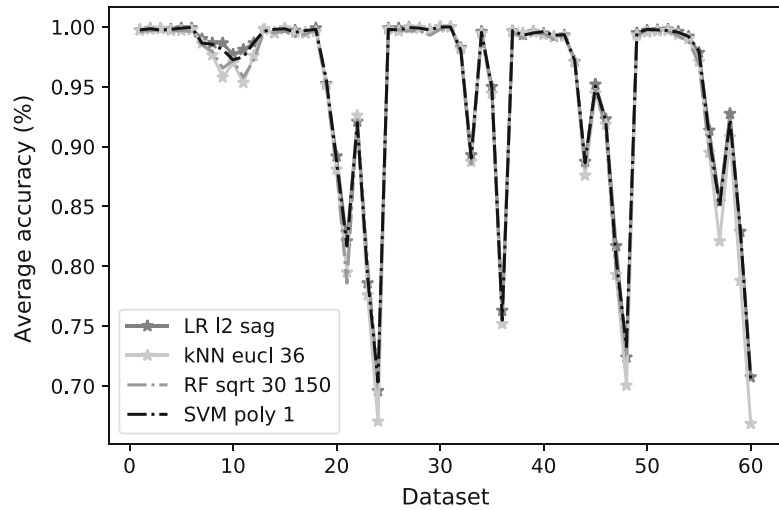


Fig. 11 Comparison of tuned classifier across all datasets

Among the classifiers tested in this survey, the kNN classifier had the worst performance among the examined algorithms, as seen in Fig. 11. Logistic regression had the best performance in the datasets with a mild bottleneck, implying that selection can be separated from neutrality linearly in mild bottlenecks. On the contrary, it showed the second worst performance, only better than kNN, in severe bottlenecks. Random forests classifier showed better performance in severe bottleneck models compared to kNN and LR. An additional advantage of RF is the ability to handle missing data, which real-world scenarios will likely include. This makes random forests a suitable classifier for selection inference. Finally, support vector machines achieved the best average performance. SVM was slightly outperformed by LR in mild bottlenecks, but achieved the best accuracy in severe bottlenecks. As a result, we suggest SVMs as the most robust classifier out of those examined in this survey.

Choosing the parameters that maximize nested k -fold CV often yields an optimistic accuracy [77]. In addition, since we use simulated data, the accuracies calculated in this report may be slightly optimistic. Still, results clearly highlight the potential of machine learning in population genetics.

In this work, we focused on a small part of the genome. Because of the advancements in sequencing technologies, whole-genome data are constantly produced, allowing to infer selection forces acting on genomes. Applying the algorithms on the genome as a whole will presumably fail to detect selection. This is due to the fact that recent selection has operated only on small parts of the whole genome, leaving the rest of it effectively neutral. Thus, if a classifier has to take a single decision for the whole genome, this will favor neutrality. A better approach is to split the whole genome into

smaller regions (sliding windows) and infer selection in each one separately. The split of the dataset into regions is performed by a sliding window algorithm that requires two parameters, the size of each window (in base pairs) and an offset that defines the starting position of the next window relative to the previous one. The pseudocode shown in Algorithm 1 describes this process.

Algorithm 1: Whole-genome selection inference in sliding windows:

```

Data: whole genome
Result: selection inference in sliding windows
read dataset;
offset = 0;
while  $start < size(dataset) - size(window)$  do
    region = dataset(start:start+offset) # take an offset-sized
    region from the genome
    outcome = prediction(region) # infer selection
    store(outcome) # store results
    start = start + offset # move to the next region
end

```

Despite the success of machine learning, it still faces challenges. First, machine learning algorithms are expensive in terms of both time and computational resources. This is a problem that will be mitigated as computer hardware and software technology advances. A general issue of the machine learning field is the dependence on quality training data. Even if an ideal algorithm would exist, it would fail to produce valuable results if the quality of the training dataset was poor. In complex problems, the need for appropriate training data that, on one hand, are labeled accurately and, on the other hand, represent correctly real scenarios is of utmost importance. Especially in population genetics, training examples are obtained via simulations because it is not possible to obtain real training example data with accurate class labeling. However, simulated data only capture a part of the evolutionary processes that may have shaped real data. Using simulated data guarantees the data quality, but it comes with the drawback of obtaining optimistic results during testing. A further approach to improve results is feature selection, which improves the quality of our data by removing noisy features. In the matter of selection inference, feature selection can improve results. If datasets contain missing or corrupted parts, then preprocessing methods exist [14].

For a most thorough study of methods related to learning, other approaches different than machine learning need to be examined as well, for example, artificial neural networks and deep learning algorithms. Currently, there are only a few studies related to deep learning in population genetics [76, 84, 103], but the

potential of the field is already apparent. Recently, a breakthrough algorithm was implemented that outcompeted both real and AI players in the strategy game Go by just knowing the rules of the game [85]. The idea of learning without human knowledge in the field of population genetics is currently far from being formulated as a proper scientific approach.

References

1. Aguadé M, Langley CH (1994) Polymorphism and divergence in regions of low recombination in *Drosophila*. In: Non-neutral evolution. Springer, Boston, pp 67–76
2. Aguade M, Miyashita N, Langley CH (1989) Reduced variation in the yellow-achaete-scute region in natural populations of *Drosophila melanogaster*. *Genetics* 122(3):607–615
3. Akey JM, Eberle MA, Rieder MJ, Carlson CS, Shriver MD, Nickerson DA, Kruglyak L (2004) Population history and natural selection shape patterns of genetic variation in 132 genes. *PLoS Biol* 2(10):e286
4. Alachiotis N, Pavlidis P (2016) Scalable linkage-disequilibrium-based selective sweep detection: a performance guide. *GigaScience* 5(1):7. <https://doi.org/10.1186/s13742-016-0114-9>
5. Alachiotis N, Pavlidis P (2018) Raised detects positive selection based on multiple signatures of a selective sweep and SNP vectors. *Commun Biol* 1(1):79
6. Alachiotis N, Pavlidis P, Stamatakis A (2012) Exploiting multi-grain parallelism for efficient selective sweep detection. In: International conference on algorithms and architectures for parallel processing. Springer, Berlin, pp 56–68
7. Alachiotis N, Stamatakis A, Pavlidis P (2012) Omegaplus: a scalable tool for rapid detection of selective sweeps in whole-genome datasets. *Bioinformatics* 28(17):2274–2275
8. Andersen EC, Gerke JP, Shapiro JA, Crissman JR, Ghosh R, Bloom JS, Félix MA, Kruglyak L (2012) Chromosome-scale selective sweeps shape *Caenorhabditis elegans* genomic diversity. *Nat Genet* 44(3):285
9. Andrew G, Gao J (2007) Scalable training of l1-regularized log-linear models. In: Proceedings of the 24th international conference on machine learning. ACM, New York, pp 33–40
10. Ansel J, Arya K, Cooperman G (2009) DMTCP: transparent checkpointing for cluster computations and the desktop. In: IEEE international symposium on parallel & distributed processing, IPDPS 2009. IEEE, Piscataway, pp 1–12
11. Arguello JR, Cardoso-Moreira M, Grenier JK, Gottipati S, Clark AG, Benton R (2016) Extensive local adaptation within the chemosensory system following *Drosophila melanogaster*'s global expansion. *Nat Commun* 7:11855 (2016)
12. Bachtrog D (2004) Evidence that positive selection drives Y-chromosome degeneration in *Drosophila miranda*. *Nat Genet* 36(5):518–522 (2004). <https://doi.org/10.1038/ng1347>
13. Barton NH (1998) The effect of hitch-hiking on neutral genealogies. *Genet Res* 72(2):123–133
14. Batista GE, Monard MC (2003) An analysis of four missing data treatment methods for supervised learning. *Appl Artif Intell* 17(5–6):519–533
15. Begun DJ, Aquadro CF (1991) Molecular population genetics of the distal portion of the x chromosome in *Drosophila*: evidence for genetic hitchhiking of the yellow-achaete region. *Genetics* 129(4):1147–1158
16. Beisswanger S, Stephan W (2008) Evidence that strong positive selection drives neofunctionalization in the tandemly duplicated polyhomeotic genes in *Drosophila*. *Proc Natl Acad Sci* 105(14):5447–5452
17. Bersaglieri T, Sabeti PC, Patterson N, Vanderploeg T, Schaffner SF, Drake JA, Rhodes M, Reich DE, Hirschhorn JN (2004) Genetic signatures of strong recent positive selection at the lactase gene. *Am J Hum Genet* 74(6):1111–1120
18. Besnier F, Kent M, Skern-Mauritzen R, Lien S, Malde K, Edvardsen RB, Taylor S, Ljungfeldt LE, Nilsen F, Glover KA (2014) Human-induced evolution caught in action: SNP-array reveals rapid amphi-atlantic spread of pesticide resistance in the salmon ectoparasite *Lepeophtheirus salmonis*. *BMC Genom* 15(1):1
19. Bigham AW, Mao X, Mei R, Brutsaert T, Wilson MJ, Julian CG, Parra EJ, Akey JM, Moore

- LG, Shriver MD (2009) Identifying positive selection candidate loci for high-altitude adaptation in Andean populations. *Hum Genom* 4(2):1
20. Bigham A, Bauchet M, Pinto D, Mao X, Akey JM, Mei R, Scherer SW, Julian CG, Wilson MJ, López Herráez D, Brutsaert T, Parra EJ, Moore LG, Shriver MD (2010) Identifying signatures of natural selection in Tibetan and Andean populations using dense genome scan data. *PLoS Genet* 6(9):e1001116. <https://doi.org/10.1371/journal.pgen.1001116>
 21. Boyd S, Parikh N, Chu E, Peleato B, Eckstein J, *et al* (2011) Distributed optimization and statistical learning via the alternating direction method of multipliers. *Found Trends Mach Learn* 3(1):1–122
 22. Braverman JM, Hudson RR, Kaplan NL, Langley CH, Stephan W (1995) The hitchhiking effect on the site frequency spectrum of DNA polymorphisms. *Genetics* 140(2):783–796
 23. Breiman L (2001) Random forests. *Mach Learn* 45(1):5–32
 24. Caicedo AL, Williamson SH, Hernandez RD, Boyko A, Fledel-Alon A, York TL, Polato NR, Olsen KM, Nielsen R, McCouch SR, *et al* (2007) Genome-wide patterns of nucleotide polymorphism in domesticated rice. *PLoS Genet* 3(9):e163
 25. Catalán A, Glaser-Schmitt A, Argyridou E, Duchon P, Parsch J (2016) An indel polymorphism in the MtnA 3' untranslated region is associated with gene expression variation and local adaptation in *Drosophila melanogaster*. *PLoS Genet* 12(4), e1005987
 26. Celine Becquet (2003) Signatures of a population bottleneck can be localised along a recombining chromosome. Tech. rep. <http://przeworski.uchicago.edu/cbecquet/MasterThesis.pdf>
 27. Charlesworth B, Morgan M, Charlesworth D (1993) The effect of deleterious mutations on neutral molecular variation. *Genetics* 134(4):1289–1303
 28. Cortes C, Vapnik V (1995) Machine learning. *Supp Vector Netw* 20:273–297
 29. Cover TM, Thomas JA (2012) Elements of information theory. Wiley, Hoboken
 30. Crisci JL, Poh YP, Mahajan S, Jensen JD (2013) The impact of equilibrium assumptions on tests of selection. *Front Genet* 4:235
 31. De A, Durrett R (2007) Stepping-stone spatial structure causes slow decay of linkage disequilibrium and shifts the site frequency spectrum. *Genetics* 176(2):969–981. <https://doi.org/10.1534/genetics.107.071464>
 32. Defazio A, Bach F, Lacoste-Julien S (2014) Saga: a fast incremental gradient method with support for non-strongly convex composite objectives. In: *Advances in neural information processing systems*, pp 1646–1654
 33. DeGiorgio M, Huber CD, Hubisz MJ, Hellmann I, Nielsen R (2016) Sweepfinder2: increased sensitivity, robustness and flexibility. *Bioinformatics* 32(12):1895–1897
 34. Depaulis F, Veuille M (1998) Neutrality tests based on the distribution of haplotypes under an infinite-site model. *Mol Biol Evol* 15(12):1788–1790
 35. Ewing GB, Jensen JD (2016) The consequences of not accounting for background selection in demographic inference. *Mol Ecol* 25(1):135–141
 36. Excoffier L, Dupanloup I, Huerta-Sánchez E, Sousa VC, Foll M (2013) Robust demographic inference from genomic and SNP data. *PLoS Genet* 9(10):e1003905
 37. Fay JC, Wu CI (2000) Hitchhiking under positive Darwinian selection. *Genetics* 155(3):1405–1413
 38. Frantz LA, Schraiber JG, Madsen O, Megens HJ, Cagan A, Bosse M, Paudel Y, Crooijmans RP, Larson G, Groenen MA (2015) Evidence of long-term gene flow and selection during domestication from analyses of Eurasian wild and domestic pig genomes. *Nat Genet* 47(10):1141–1148
 39. Friedman J, Hastie T, Tibshirani R (2001) The elements of statistical learning. Springer series in statistics New York, vol 1. Springer, Berlin
 40. Gerbault P, Powell A, Thomas MG (2012) Evaluating demographic models for goat domestication using mtDNA sequences. *Anthropozoologica* 47(2):64–76
 41. Glinka S, Ometto L, Mousset S, Stephan W, De Lorenzo D (2003) Demography and natural selection have shaped genetic variation in *Drosophila melanogaster*: a multi-locus approach. *Genetics* 165(3):1269–1278
 42. Hartl DL, Clark AG, Clark AG (1997) Principles of population genetics, vol 116. Sinauer Associates, Sunderland
 43. Hawkins DM (2004) The problem of overfitting. *J Chem Inf Comput Sci* 44(1):1–12
 44. Hernandez RD (2008) A flexible forward simulator for populations subject to selection and demography. *Bioinformatics* 24(23):2786–2787

45. Hudson RR (2002) Generating samples under a Wright-Fisher neutral model of genetic variation. *Bioinformatics* 18 (2):337–338
46. Hudson RR, Slatkin M, Maddison W (1992) Estimation of levels of gene flow from DNA sequence data. *Genetics* 132(2), 583–589
47. Innan H, Stephan W (2003) Distinguishing the hitchhiking and background selection models. *Genetics* 165(4):2307–2312 (2003)
48. Johnson R, Zhang T (2013) Accelerating stochastic gradient descent using predictive variance reduction. In: *Advances in neural information processing systems*, pp 315–323
49. Kaplan NL, Hudson R, Langley C (1989) The “hitchhiking effect” revisited. *Genetics* 123(4):887–899
50. Kern AD, Schrider DR (2018) diploS/HIC: an updated approach to classifying selective sweeps. *Genes Genom Genet* 8:1959–1970
51. Kim Y, Nielsen R (2004) Linkage disequilibrium as a signature of selective sweeps. *Genetics* 167(3):1513–1524. <https://doi.org/10.1534/genetics.103.025387>
52. Kim Y, Stephan W (2002) Detecting a local signature of genetic hitchhiking along a recombining chromosome. *Genetics* 160 (2):765–777
53. Langley CH, MacDonald J, Miyashita N, Aguade M (1993) Lack of correlation between interspecific divergence and intraspecific polymorphism at the suppressor of forked region in *Drosophila melanogaster* and *Drosophila simulans*. *Proc. Natl. Acad. Sci.* 90(5):1800–1803
54. Laurent S, Pfeifer SP, Settles ML, Hunter SS, Hardwick KM, Ormond L, Sousa VC, Jensen JD, Rosenblum EB (2016) The population genomics of rapid adaptation: disentangling signatures of selection and demography in white sands lizards. *Mol Ecol* 25(1):306–323
55. Li H (2011) A new test for detecting recent positive selection that is free from the confounding impacts of demography. *Mol Biol Evol* 28(1):365–375. <https://doi.org/10.1093/molbev/msq211>
56. Li H, Stephan W (2006) Inferring the demographic history and rate of adaptive substitution in *Drosophila*. *PLoS Genet* 2(10):e166
57. Lin K, Li H, Schlötterer C, Futschik A (2011) Distinguishing positive selection from neutral evolution: boosting the performance of summary statistics. *Genetics* 187(1):229–44. <https://doi.org/10.1534/genetics.110.122614>
58. Luo Q, Ahmad K, Fu HY, Wang JD, Chen RK, Gao SJ (2016) Genetic diversity and population structure of Sorghum mosaic virus infecting *Saccharum* spp. hybrids. *Ann. Appl. Biol.* 169(3):398–407
59. Martin SH, Möst M, Palmer WJ, Salazar C, McMillan WO, Jiggins FM, Jiggins CD (2016) Natural selection and genetic diversity in the butterfly *Heliconius melpomene*. *Genetics* 203(1):525–541
60. McManus KF, Kelley JL, Song S, Veeramah KR, Woerner AE, Steverson LS, Ryder OA, Kidd JM, Wall JD, Bustamante CD, *et al* (2015) Inference of gorilla demographic and selective history from whole-genome sequence data. *Mol Biol Evol* 32(3):600–612
61. Michalski RS, Carbonell JG, Mitchell TM (2013) *Machine learning: an artificial intelligence approach*. Springer, New York
62. Miyashita NT (1990) Molecular and phenotypic variation of the *Zw* locus region in *Drosophila melanogaster*. *Genetics* 125 (2):407–419
63. Montano V, Didelot X, Foll M, Linz B, Reinhardt R, Suerbaum S, Moodley Y, Jensen JD (2015) Worldwide population structure, long-term demography, and local adaptation of *helicobacter pylori*. *Genetics* 200 (3):947–963
64. Nasrabadi NM (2007) Pattern recognition and machine learning. *J Electron Imag* 16 (4):049901
65. Nielsen R, Williamson S, Kim Y, Hubisz MJ, Clark AG, Bustamante C (2005) Genomic scans for selective sweeps using SNP data. *Genom Res* 15(11):1566–1575
66. Alpaydin E (2009) *Introduction to machine learning*. MIT press (Cambridge)
67. Orengo DJ, Aguadé M (2004) Detecting the footprint of positive selection in a European population of *Drosophila melanogaster*: multilocus pattern of variation and distance to coding regions. *Genetics* 167 (4):1759–1766. <https://doi.org/10.1534/genetics.104.028969>
68. Papadantonakis S, Poirazi P, Pavlidis P (2016) CoMuS: simulating coalescent histories and polymorphic data from multiple species. *Mol Ecol Resour* 16(6):1435–1448
69. Pavlidis P, Alachiotis N (2017) A survey of methods and tools to detect recent and strong positive selection. *J Biol Res-Thessaloniki* 24 (1):7
70. Pavlidis P, Hutter S, Stephan W (2008) A population genomic approach to map recent positive selection in model species. *Mol Ecol* 17(16):3585–3598

71. Pavlidis P, Jensen JD, Stephan W (2010) Searching for footprints of positive selection in whole-genome SNP data from nonequilibrium populations. *Genetics* 185(3):907–922
72. Pavlidis P, Jensen JD, Stephan W, Stamatakis A (2012) A critical assessment of storytelling: gene ontology categories and the importance of validating genomic scans. *Mol Biol Evol* 29(10):3237–3248. <https://doi.org/10.1093/molbev/mss136>
73. Pavlidis P, Živković D, Stamatakis A, Alachiotis N (2013) SweeD: likelihood-based detection of selective sweeps in thousands of genomes. *Mol Biol Evol* 30(9):2224–2234
74. Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, Blondel M, Prettenhofer P, Weiss R, Dubourg V, *et al* (2011) Scikit-learn: machine learning in python. *J Mach Learn Res* 12:2825–2830
75. Pickrell JK, Coop G, Novembre J, Kudaravalli S, Li JZ, Absher D, Srinivasan BS, Barsh GS, Myers RM, Feldman MW, *et al* (2009) Signals of recent positive selection in a worldwide sample of human populations. *Genom Res* 19(5):826–837
76. Quang D, Xie X (2016) DanQ: a hybrid convolutional and recurrent deep neural network for quantifying the function of DNA sequences. *Nucleic Acids Res* 44(11), e107–e107
77. Refaëilzadeh P, Tang L, Liu H (2009) Cross-validation. In: *Encyclopedia of database systems*. Springer, Berlin, pp 532–538
78. Rubin CJ, Zody MC, Eriksson J, Meadows JR, Sherwood E, Webster MT, Jiang L, Ingman M, Sharpe T, Ka S, *et al* (2010) Whole-genome resequencing reveals loci under selection during chicken domestication. *Nature* 464(7288):587
79. Sabeti PC, Schaffner SF, Fry B, Lohmueller J, Varilly P, Shamovsky O, Palma A, Mikkelsen T, Altshuler D, Lander E (2006) Positive natural selection in the human lineage. *Science* 312(5780):1614–1620
80. Schlenke TA, Begun DJ (2004) Strong selective sweep associated with a transposon insertion in *Drosophila simulans*. *Proc Natl Acad Sci USA* 101(6):1626–1631
81. Scholkopf B, Smola AJ (2001) *Learning with kernels: support vector machines, regularization, optimization, and beyond*. MIT Press, Cambridge (2001)
82. Schrider DR, Kern AD (2016) S/HIC: robust identification of soft and hard sweeps using machine learning. *PLoS Genet* 12(3): e1005928
83. Schrider DR, Kern AD (2018) Supervised machine learning for population genetics: a new paradigm. *Trends Genet* 34(4):301–312
84. Sheehan S, Song, YS (2016) Deep learning for population genetic inference. *PLoS Comput Biol* 12(3):e1004845
85. Silver D, Schrittwieser J, Simonyan K, Antonoglou I, Huang A, Guez A, Hubert T, Baker L, Lai M, Bolton A, *et al* (2017) Mastering the game of go without human knowledge. *Nature* 550(7676):354
86. Smith JM, Haigh J (1974) The hitch-hiking effect of a favourable gene. *Genet Res* 23(1):23–35
87. Städler T, Haubold B, Merino C, Stephan W, Pfäffelhuber P (2009) The impact of sampling schemes on the site frequency spectrum in nonequilibrium subdivided populations. *Genetics* 182(1):205–216
88. Stephan W (2010) Genetic hitchhiking versus background selection: the controversy and its implications. *Philos Trans R Soc Lond Ser B Biol Sci* 365(1544):1245–1253. <https://doi.org/10.1098/rstb.2009.0278>
89. Stephan W, Langley CH (1989) Molecular genetic variation in the centromeric region of the x chromosome in three *Drosophila ananassae* populations. i. contrasts between the vermilion and forked loci. *Genetics* 121(1):89–99 (1989)
90. Stephan W, Wiehe THE, Lenz MW (1992) The effect of strongly selected substitutions on neutral polymorphism: analytical results based on diffusion theory. *Theor. Popul Biol* 41(2):237–254. [https://doi.org/10.1016/0040-5809\(92\)90045-U](https://doi.org/10.1016/0040-5809(92)90045-U)
91. Stephan W, Song YS, Langley CH (2006) The hitchhiking effect on linkage disequilibrium between linked neutral sites. *Genetics* 172(4):2647–2663
92. Tajima, F (1989) Statistical method for testing the neutral mutation hypothesis by DNA polymorphism. *Genetics* 123(3):585–595
93. Thornton KR, Jensen JD (2007) Controlling the false positive rate in multilocus genome scans for selection. *Genetics* 175(2):737–750
94. Trujillo JT, Beilstein MA, Mosher RA (2016) The Argonaute-binding platform of NRPE1 evolves through modulation of intrinsically disordered repeats. *New Phytol* 212(4):1094–1105. <https://doi.org/10.1111/nph.14089>
95. Voigt S, Laurent S, Litovchenko M, Stephan W (2015) Positive selection at the polyhomeotic locus led to decreased thermosensitivity of gene expression in temperate

- Drosophila melanogaster*. *Genetics* 200 (2):591–599
96. Wall JD (1999) Recombination and the power of statistical tests of neutrality. *Genet Res* 74(1):65–79
 97. Wang MS, Zhang RW, Su LY, Li Y, Peng MS, Liu HQ, Zeng L, Irwin DM, Du JL, Yao YG, *et al* (2016) Positive selection rather than relaxation of functional constraint drives the evolution of vision during chicken domestication. *Cell Res* 26(5):556
 98. Wilches R, Voigt S, Duchon P, Laurent S, Stephan W (2014) Fine-mapping and selective sweep analysis of QTL for cold tolerance in *Drosophila melanogaster*. *G3 Genes Genom Genet* 4(9):1635–1645
 99. Yang SY, Han MJ, Kang LF, Li ZW, Shen YH, Zhang Z (2014) Demographic history and gene flow during silkworm domestication. *BMC Evol Biol* 14(1):185
 100. Yuan Y, Zhang Q, Zeng S, Gu L, Si W, Zhang X, Tian D, Yang S, Wang L (2017) Selective sweep with significant positive selection serves as the driving force for the differentiation of japonica and indica rice cultivars. *BMC Genom* 18(1):307
 101. Zeng K, Fu YX, Shi S, Wu CI (2006) Statistical tests for detecting positive selection by utilizing high-frequency variants. *Genetics* 174(3):1431–1439
 102. Zhang Z, Jia Y, Almeida P, Mank JE, van Tuinen M, Wang Q, Jiang Z, Chen Y, Zhan K, Hou S, *et al* (2018) Whole-genome resequencing reveals signatures of selection and timing of duck domestication. *Giga-Science* 7(4):giy027
 103. Zhou J, Troyanskaya OG (2015) Predicting effects of noncoding variants with deep learning-based sequence model. *Nat Methods* 12(10):931
 104. Živković D, Stephan W (2011) Analytical results on the neutral non-equilibrium allele frequency spectrum based on diffusion theory. *Theor Popul Biol* 79(4):184–191

Open Access This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.

