HHS Public Access

Author manuscript

Trends Genet. Author manuscript; available in PMC 2018 April 18.

Published in final edited form as:

Trends Genet. 2018 April ; 34(4): 301–312. doi:10.1016/j.tig.2017.12.005.

Supervised Machine Learning for Population Genetics: A New Paradigm

Daniel R. Schrider[1,*] and **Andrew D. Kern**[1,*]

[1]Department of Genetics, and Human Genetics Institute of New Jersey, Rutgers University, Piscataway, NJ 08554, USA

Abstract

As population genomic datasets grow in size, researchers are faced with the daunting task of making sense of a flood of information. To keep pace with this explosion of data, computational methodologies for population genetic inference are rapidly being developed to best utilize genomic sequence data. In this review we discuss a new paradigm that has emerged in computational population genomics: that of supervised machine learning (ML). We review the fundamentals of ML, discuss recent applications of supervised ML to population genetics that outperform competing methods, and describe promising future directions in this area. Ultimately, we argue that supervised ML is an important and underutilized tool that has considerable potential for the world of evolutionary genomics.

Machine Learning for Population Genetics

Population genetics over the past 50 years has been squarely focused on reconciling molecular genetic data with theoretical models that describe patterns of variation produced by a combination of evolutionary forces. This interplay between empiricism and theory means that many advances in the field have come from the introduction of new stochastic population genetic models, often of increasing complexity, that describe how population parameters (e.g., recombination or mutation rates) might generate specific features of genetic polymorphism (e.g., the **site frequency spectrum**, SFS; see Glossary). The goal, broadly stated, is to formulate a model that describes how nature will produce patterns of variation that we observe. With such a model in hand, all one would need to do would be to estimate its parameters, and in so doing learn everything about the evolution of a given population.

Thus an overwhelming majority of population genetics research has focused on classical statistical estimation from a convenient probabilistic model (i.e., the Wright–Fisher model), or through an approximation to that model (i.e., the coalescent). The central assertion here is that the model sufficiently describes the data such that insights into nature can be made through parameter estimation. This mode of analysis that pervades population genetics is what Leo Breiman [1] famously referred to as the ‘data modeling culture’, wherein

*Correspondence: dan.schrider@rutgers.edu (D.R. Schrider) and kern@biology.rutgers.edu (A.D. Kern).

independent variables (i.e., the evolutionary and genomic parameters) are fed into a model and the response variables (some aspect of genetic variation) come out the other side. Models are validated in this worldview through the use of goodness-of-fit tests or examination of residuals (a recent modern example can be found in [2]).

In this review we argue that researchers should consider utilizing a powerful mode of analysis that has recently emerged within population genetics – the 'algorithmic modeling culture', or what is now commonly called machine learning (ML). Over the past decade ML methods have revolutionized entire fields, including speech recognition [3], natural language processing [4], image **classification** [5], and bioinformatics [6–8]. However, the application of ML to problems in population and evolutionary genetics is still in its infancy, except for a few examples [9–18]. ML approaches have several desirable features, and perhaps foremost among them is their potential to be agnostic about the process that creates a given dataset. ML, as a field, aims to optimize the predictive accuracy of an algorithm rather than perform parameter estimation of a probabilistic model. What this means in practice is that ML methods can teach us something about nature, even if our models used to describe nature are imprecise. An equally important advantage of the ML paradigm is that it enables the efficient use of high-dimensional inputs which act as dependent variables, without specific knowledge of the joint probability distribution of these variables. Inputs that consist of thousands of variables (also known as 'features' in the ML world) have been used with great success (e.g., [19,20]), and increases in the number of features can often yield greater predictive power [1]. Given the ever-increasing dimensionality of modern genomic data, this is a particularly desirable property of ML. In this paper we describe several examples where, through a hybrid of the 'data modeling' and 'algorithmic modeling' paradigms, ML methods can leverage high-dimensional data to attain far greater predictive power than competing methods. These early successes demonstrate that ML approaches could have the potential to revolutionize the practice of population genetic data analysis.

An Introduction to Machine Learning

ML is generally divided into two major categories (although hybrid strategies exist): supervised learning [21] and unsupervised learning [22]. Unsupervised learning is concerned with uncovering structure within a dataset without prior knowledge of how the data are organized (e.g., identifying clusters). A familiar example of unsupervised learning is principal component analysis (PCA), which in the context of population genetics is used for discovering unknown relatedness relationships among individuals. PCA takes as input a matrix of genotypes (often of very high dimensionality) and then produces a lower-dimensional summary that can reveal how genotypes cluster. An excellent example of the application of PCA to population genetics can be found in Novembre *et al.* [23] where PCA was used to show how relationships among individuals sampled from Europe largely mirrored geography. Supervised learning, by contrast, relies on prior knowledge about an example dataset to make predictions about new datapoints. Generally, supervised ML 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, to **train** the predictor (Boxes 1 and 2).

Box 1

Supervised Learning in Cartoon Form

Perhaps the simplest way to understand supervised ML is graphically. Imagine a scenario in which we wish to train a computer to differentiate between two kinds of fruit, for examples apples and oranges, on the basis of two measurements (x_1 and x_2) taken from each example (Figure I). In supervised ML we will use known, labeled examples, in other words a 'training set' (the filled-in datapoints in Figure I) to learn a function that can discriminate between our data classes. Once we have 'learned' this function we can then use our trained oracle to predict class membership of new, unlabeled examples (the unfilled datapoints in Figure I).

Box 2

Supervised Learning in Draft Form

Supervised ML approaches algorithmically create from a given dataset a function that takes as input a vector and then emits a predicted value for each datapoint. More formally, these methods learn a function, f, that predicts a response variable, y, from a feature vector, x, containing M input variables, such that $f(x) = y$. If y is a categorical variable, we refer to the task as a **classification** problem, whereas if y is a continuous variable we refer to it as **regression**. In supervised learning, the objective is to optimize $f: x \rightarrow y$ using a 'training set' of labeled data (i.e., whose response values are known). That is, we assume we have a set of training data of length n of the form $\{(x_1, y_1), ..., (x_n, y_n)\}$, where $x \in \mathbf{R}^M$. A variety of learning algorithms exist which can create functions that can perform either classification or regression, including support vector machines (SVMs [80]), **decision trees** [81], random forests [50], boosting [82], and artificial neural networks (ANNs [83]) which in modern form are subsumed under the umbrella of deep learning [84]. These algorithms differ in how they structure and train f (see brief descriptions in the Glossary).

To proceed with building f we must define a **loss function**, L, that indicates how good or bad a given prediction is. A simple choice for a loss function in the context of classification would be the indicator function such that $L(f(x), y) = 1(f(x) \neq y)$. For regression one might consider the squared deviation $L(f(x), y) = (f(x) - y)^2$. Finally, we define the **risk function** which is typically the average value of L across the training set. Training is the process of minimizing this risk function.

Once training is complete, we must evaluate our performance on an independent **test set**. This step allows one to assess whether f has become sensitive to the general characteristics of the problem at hand, rather than to characteristics particular to data examples in the training set (what is known as **overfitting**). For **binary classification** we might characterize the false positive and false negative rates or related measures such as **precision** and **recall**. A particularly helpful construct in the case of multiclass classification is the **confusion matrix**, which is simply the contingency table of true versus predicted class labels for each class. For regression, one could use any tool for

evaluating model fit (e.g., R^2) or examine the distribution of values of one or more loss functions. Residuals can also be checked for evidence of bias to anticipate which types of data are likely to produce erroneous predictions.

There have been a multitude of important applications of unsupervised ML in evolutionary genomics beyond PCA. One popular methodology that has been wildly successful in population and evolutionary genetics is hidden Markov models (HMMs [24]). HMMs are a class of probabilistic graphical model that are well suited to segmenting data that appears as a linear sequence, such a chromosomes. For instance, with phylogenetic data HMMs have been used to uncover differences in evolutionary rates along a chromosome [25,26]. Furthermore, HMMs have been used to infer how the phylogeny itself changes across chromosomes as a result of recombination [10,27,28]. In the context of population genetic data HMMs have been leveraged to detect regions of the genome under positive or negative selection [11], as well as to localize selective sweeps [12,29].

Although unsupervised ML has been deployed widely and effectively throughout the field, to date less attention has been paid to supervised learning. We give here a brief overview of the paradigm of supervised ML and highlight recent population genetic studies leveraging these approaches.

Why use Machine Learning?

Our basic description of supervised ML approaches in Box 2 demonstrates their central rationale: ML focuses on algorithmically constructed models with optimal prediction as their goal rather than parametric data modeling. Furthermore, ML offers several advantages in addition to accurate prediction. Perhaps most important among them is the ability to circumvent using idealized, parametric models of the data when labeled training data can be obtained from empirical observation (an example of this scenario is given in the following section). Indeed, in such cases we can use ML to train algorithms to recognize phenomena as they are in nature, rather than how we choose to represent them in a model. Further, in cases where empirically derived training sets are not available, simulation can be used to generate training sets. This ability to use simulation as a stand-in for observed data is key for population genetics applications, where adequately sized datasets with high-confidence labels are currently hard to obtain. Of course, using simulation for training obviates the model agnosticism that is attractive about ML in the first place, and thus in using simulation to generate training sets one must be concerned with issues of model mis-specification exactly as when working with traditional, generative models. While that is so, discriminative ML models have been shown to be more robust to model mis-specification than traditional data models [30].

Even when empirical training data cannot feasibly be obtained, there are notable advantages of supervised ML methods. Most importantly, these methods are specifically geared toward using high-dimensional data as the input. Typically, classical statistical methods suffer from what has been called the ‘curse of dimensionality’ whereby high-dimensional data become sparse and thus very difficult to fit models to. By contrast, most supervised ML methods perform better when the input data have a large number of features, in what is commonly

called the ‘blessing of dimensionality’ (e.g., [1,31]). A good example of this comes from the highly cited work of Amit and Geman [19] on using a **random forest**-like procedure for handwriting recognition: it took as input a **feature vector** containing thousands of variables, and proved to be highly accurate. In a more modern setting, **deep learning** methods have been shown both theoretically and in practice to be able to circumvent the curse of dimensionality in many settings [32,33]. This attribute lends significant strength to population genetics analysis: while inferences are traditionally based on a single summary statistic devised for the given task (e.g., [34–40]), below we describe several recent studies which demonstrate that far greater statistical power can be achieved by simultaneously examining multiple aspects of genetic variation across the genome. Importantly, many ML methods offer direct ways to assess which features of the input are driving inferences, information which can yield insights about the underlying processes [1].

The last benefit we wish to touch upon is computational efficiency. While training of supervised ML algorithms is computationally costly – especially if simulation is used for the training set – once an algorithm is trained, prediction from it is exceedingly fast even in situations where a large number of predictions is required (e.g., genome-wide scans). This means that there will be an upfront cost to training (typically hours or days), but genome-wide inference proceeds rapidly thereafter. Moreover, because many ML approaches (e.g., deep learning) have the ability to generalize beyond their input parameters (e.g., [41]), training sets can be considerably smaller than those used by approaches such as approximate Bayesian computation (ABC [42]).

Supervised ML in Population Genetics by Training on Real Data: Finding Purifying Selection

When empirically derived training data are available, supervised ML can be used to make accurate predictions in datasets that cannot be adequately modeled with a reasonable number of parameters. For instance, a current goal in modern genomics is to be able to predict functional regions of the genome using bioinformatics techniques. While there are numerous sources of information to leverage for this problem, including comparative [26] and functional genomics [43], the best manner in which to incorporate population genomic variation to aid in these predictions is a matter of active research. Toward this end a supervised ML approach was recently used to discriminate between genomic regions experiencing purifying selection and those free from selective constraint on the basis of population genomic data alone [16]. In this study a **support vector machine** (SVM) was used that employed as its input the SFS from all 1092 individuals from the Phase I release of **1000 Genomes Dataset** which consisted of 14 population samples from diverse global locations [44]. Had this been done using all these data simultaneously in a ‘classical’ population genetics setting the researchers would have been forced to fit a demographic model that described the joint divergence and population size changes of all 14 population samples, a daunting task indeed. While the SFS is well known to be affected by demography as well as by selection [45], by constructing a training set of regions experiencing purifying selection (inferred from a phylogenetic comparison of non-human mammals) the intractable problem of modeling the joint demographic history of the dataset was able to effectively be

sidestepped. An SVM was thus able to be trained and tested using empirical data, achieving ~88% accuracy [16].

By comparing the predictions from this classifier, which reveal purifying selection occurring in recent evolutionary history, with phylogenetic signatures of more ancient selection, regions showing evidence of functional turnover in the human genome were able to be identified. These candidate regions were found to be highly enriched in the regulatory domains of genes important for proper central nervous system development. Moreover, another study [46] recently found that the presence of these candidate regions near a gene was more predictive of human-specific changes of expression in the brain than was the presence of well-known human-accelerated regions identified from interspecific comparisons [47]. This result lends credence both to our own predictions and more generally to the utility of supervised ML approaches in evolutionary genetics.

Finding Selective Sweeps in the Genome

One population genetic question that has received recent attention using ML approaches is that of detecting selective sweeps: the signature left by an adaptive mutation that rapidly increases in allele frequency until reaching fixation [48]. While the classical population genetic strategy for finding sweeps has been to carefully devise test statistics sensitive to selective perturbations [34–40], in recent years several groups have begun leveraging combinations of statistics through supervised ML to improve inferential power. While each of these methods differ in the exact combination of summary statistics used, their unifying feature is that training sets are generated using coalescent simulations with and without selective sweeps. The first of these studies [13] used a SVM to combine the ω statistic of Kim and Nielsen (which measures the spatial pattern of LD expected around a sweep [38]) with composite-likelihood ratio of Nielsen *et al.* (also known as CLR, which highlights the spatial skew in the SFS expected around a sweep [49]). They found that these two statistics in concert had greater power to detect sweeps. Another study [15] took the approach of encoding the SFS as the feature vector (i.e., each bin in the SFS is one feature), and then used an SVM to discriminate between selective sweeps and neutrality. Others [9] have used **boosting** to identify sweeps on the basis of a feature vector containing six different summary statistics each measured across several genomic subwindows surrounding the focal window. In a related effort, a series of boosting classifiers were recently used to detect selective sweeps and classify them according to whether they have reached fixation (complete vs incomplete) as well as by their timing (recent vs ancient) [14]. Finally, S/HIC (soft/hard inference through classification), which uses a variant of a random forest [50] called an extra-trees classifier [51] to detect both classic **hard sweeps** from *de novo* mutations and **soft sweeps** resulting from selection on previously segregating variants [52,53], was recently reported [17]. As described in Box 3, S/HIC is able to detect sweeps with high sensitivity and specificity even in the face of non-equilibrium demography which confounds many other methods. The success of S/HIC and the other efforts listed above demonstrates that an appropriately designed ML approach can make rapid advances in performance on difficult problems that have received attention for decades.

Box 3

A Closer Look at S/HIC

S/HIC [17] uses a feature vector designed to be not only sensitive to hard and soft sweeps but also robust to the confounding effects of both linked positive selection (i.e., 'soft shoulders' [85]) and non-equilibrium demography [45,86]. This feature vector included values of nine different statistics that were each measured in several adjacent subwindows (Figure III) in a similar vein to the evolBoosting of Lin *et al.* [9]. What set this feature vector apart is that, for each statistic, the value in each subwindow was normalized by dividing by the sum across all subwindows. Thus, the true value of a given statistic in a given subwindow is ignored, while the relative values across the larger window are examined. The reasoning behind this choice is that, although demographic events may affect values of population genetic summaries genome-wide (which S/HIC ignores), selective sweeps may result in more dramatic localized skews in these statistics (which S/HIC captures). The results of this design are impressive: S/HIC is able to detect sweeps under challenging demographic scenarios, often with no loss in power even when the demographic history is grossly mis-specified during training (e.g., if there is an unknown population bottleneck), a scenario which catastrophically compromises many other methods [17,87]. Thus, ML methods – especially those with appropriately designed feature vectors – can be robust to modeling choices even when training data are simulated.

In Figure III we illustrate the S/HIC classification strategy and the values included in its feature vector. This figure demonstrates how much additional information S/HIC utilizes in making its predictions in comparison to more traditional population genetic tests, especially those relying on a single statistic. In particular, the S/HIC feature vector not only includes multiple statistics, each of which is designed to capture different aspects of genealogies, but also how these statistics vary along the chromosome. In addition to greater robustness to demography as discussed above, incorporating all of this information yields greater discriminatory power, and for this reason such multidimensional methods will be preferable to univariate approaches. We recently applied S/HIC to six human populations with complex demographic histories, where it revealed that soft sweeps appear to account for the majority of recent adaptive events in humans [88]; the success of this analysis demonstrates the practicality of applying such ML strategies to real data.

The methods listed above have two commonalities: they use ML to perform classification on multidimensional input, and they handily outperform more traditional univariate methods. However, these methods also differ from one another substantially in several facets: the particular ML framework used, the makeup of the feature vector, and the types of sweeps they seek to detect. Thus, the success of these methods underscores not only the power but also remarkable flexibility of supervised ML. By working within the supervised ML paradigm one can effectively tailor a predictor to whatever task is at hand simply by altering the construction of the feature vector and training dataset, and in so doing make more detailed predictions than is possible using a single statistic.

Unlike the problem of detecting purifying selection, for which a training set may be constructed, we lack an adequate number of selective sweeps whose parameters are known precisely (e.g., the time of the sweep, strength of selection). Thus, the studies described above used simulation to generate training sets. The general idea is to simulate data from one or several population genetic models in which parameters are either specified precisely or defined by prior distributions, use those data to train an ML algorithm, and then perform either classification or **regression** (i.e., parameter estimation). In this context supervised ML allows for likelihood-free inference of population genetic models similar in spirit to ABC. Although, like ABC, this approach requires modeling assumptions, it nonetheless offers numerous advantages as described in Box 4 where we contrast ABC with supervised ML.

Box 4

Comparing Supervised ML and ABC for Population Genetic Inference

Using supervised ML with training data simulated from a specified set of population genetic models is similar in spirit to approximate Bayesian computation (ABC), except for some notable distinctions. ABC begins by simulating a large number of examples whose model parameters are drawn from prior distributions, and then summarizes these simulations with vectors of population genetic summary statistics. Next, in 'classical' ABC, only those simulations most similar to the observed dataset are retained – a process known as rejection sampling – to approximate the probability distribution for each parameter value given the observed data. ABC is easy to implement, flexible, and has been proven effective in several scenarios. However, ABC has some important drawbacks that ML overcomes. Most importantly, when using large feature vectors, ABC is susceptible to the curse of dimensionality [59] – much effort has therefore gone into dimensionality reduction and feature selection for ABC (reviewed in [89]). While this is so, reducing dimensionality might lead to loss of information if the remaining summaries are not sufficient statistics of the data. This contrasts with modern ML algorithms which can benefit from high-dimensional data rather that suffer from them.

A second drawback of classical ABC is its computational burden. Although both ML and ABC require a large number of simulations, ABC does not make efficient use of all of this computation because it typically depends on rejection sampling. Work has been done to retain more of the simulations in ABC, for instance by weighing their influence on parameter estimation according to their similarity to the observed data [62]. However, ML methods naturally use all of the simulations to learn the mapping of data to parameters. Further, deep learning methods have the potential to generalize non-locally [32], allowing them to make accurate predictions for data very different from those in the training set. For these reasons, ML may require considerably fewer simulations than ABC. Furthermore, ML methods need not re-examine these simulations to perform downstream prediction, unlike ABC, and thus further inference is very fast.

A third difference between ML and ABC is that of interpretability. In the realm of ABC it is not clear which summaries are responsible for a signal. By contrast, many ML methods allow direct measurement of the contribution of each feature. Thus, despite their use of algorithmically generated models, ML algorithms are far from black boxes.

Finally, it is important to note that newer versions of ABC, that do not depend on rejection sampling, are often simply examples supervised ML approaches at their core [62,63,90], thus to some large degree the dichotomy pointed to above is destined to become moot.

Inferring Demography and Recombination

Another emerging use of supervised ML in population genetics has been for inference of demographic history and recombination rates. Indeed, much attention in the field has been placed on developing methods for the inference of population size histories and patterns of population splitting and migration [54–58]. ABC methods are among the most popular for inferring demographic histories [59]. Interestingly, several groups have experimented with augmenting ABC by using ML for selecting the optimal combination of summary statistics [60] or even generating them [61]. While this is a promising direction for feature engineering, others have directly used ML to estimate posterior distributions of demographic parameters. For instance, Blum and François [62] used a feed-forward **artificial neural network** (ANN) to learn the mapping of summary statistics onto parameters with excellent results, particularly with respect to computational cost savings.

In addition to demographic parameter estimation, supervised ML has been used recently for demographic model selection (a possibility pointed to by Blum and François). For instance, it was recently shown [63] that random forests outperform ABC in both accuracy and computational cost when performing demographic model selection, together with greater robustness to the choice of summary statistics included in the input vector. In a recent preprint [64], Extra-Trees classifiers were applied to a problem of locus-specific demographic model selection: that of identifying regions with gene flow between a pair of closely related species with far greater accuracy than previous methods. Thus in general, ML methods show great promise in demographic estimation and model selection, and may soon be the preferred choice over ABC.

Supervised ML has also been applied to characterize the rates and patterns of recombination in the genome. This work has again been done with or without simulation of training data. For instance, a random forest classifier was trained to distinguish among recombination rate classes on the basis of sequence motifs to show that such motifs are predictive of recombination rate in *Drosophila melanogaster* [65]. This work used annotated rates of recombination based on a classical population genetics estimator to define the training set. By contrast, methodology has been developed [66,67] that uses boosting to infer recombination rate maps from large sample sizes on the basis of simulated training data. The latest method, FastEPRR (fast estimation of population recombination rates), has much greater computational efficiency than, and equal accuracy to, the widely used LDhat [68]. Although application of supervised ML methods to this problem has begun only recently, the success of FastEPRR suggests the potential of future gains using these approaches.

Coestimation of Selection and Demography

It is well known that demographic events can mimic the effects of selection [45], and conversely that selection can confound demographic estimation [69,70]. This implies that, although one can attempt to design more robust approaches (e.g., S/HIC, discussed above), the ideal strategy would be to simultaneously make inferences about both of these phenomena. How then can one perform coestimation of parameters related to multiple evolutionary phenomena? A promising approach that utilizes supervised ML, in this case deep learning, was recently introduced by [18]. Here a deep neural network, called evoNet, was developed to simultaneously infer population size changes in a three-epoch model and detect hard and soft selective sweeps as well as regions under balancing selection. What makes this research particularly important is that using this method the researchers were able to perform simultaneous classification of loci into selective classes and demographic parameter estimation (based on averages estimated over loci classified as neutral) through the use of a neural network architecture that outputs both categorical and continuous parameters. This inherent flexibility of ML, and deep learning architectures in particular, opens up a whole slew of opportunities for doing population genomic inference in ways that have never before been possible (discussed below).

Concluding Remarks and Future Directions

The future of population genomic analysis rests in our ability to make sense of large and ever-growing datasets. Toward this end, supervised ML techniques represent a new paradigm for analysis, one uniquely suited for making inferences in the context of high-dimensional data produced by an unknown or imprecisely parameterized model. We have reviewed here a selection of early applications of supervised ML tools to population genomic data. The overwhelming take-home message is that supervised ML provides robust, computationally efficient inference for several problems that are difficult to gain traction on via classical statistical approaches.

We believe that population genetics is now poised for an explosion in the use of supervised ML approaches. Deep learning in particular, with its incredibly flexible input and output structure, should be an important area of future research, and its earliest application [18] has yielded the crucial ability to coestimate selection and demography, a central goal of population genetics analysis over the past 15 years. Indeed, deep learning could potentially alter the way that we even think about the nature of our input data itself. For example, one flavor of deep learning, convolutional neural networks (CNNs), have made astounding advances in our ability to learn parameters from image data [71]. Rather than learning on population genetic summary statistics calculated from a multiple sequence alignment (e.g., [9,17]), one could instead treat an image of the alignment itself as the input. While these data would be extremely high-dimensional, the structure of CNNs allows them to implicitly perform dimensionality reduction while capturing salient structures in the input data [72], allowing accurate and efficient classification and regression (additional possible future avenues of ML in population genetics are listed in the Outstanding Questions). While these are exciting prospects, a general challenge lies ahead in making more structured population genetics inferences beyond simple parameter estimation or classification. For instance, it is

not clear to what extent the supervised ML techniques discussed above could be used to infer genealogies or other tree-like structures (but see [73]). In general, however, the current explosion in deep learning research promises future improvements in our ability to make evolutionary inferences well beyond current capabilities; the challenge for population geneticists then is to adapt such methods for our own uses.

Outstanding Questions

While a few comparisons have shown that ML can outperform ABC, a more thorough assessment of the strengths and limitations of each approach across a variety of problems (e.g., on simulated data) is warranted. In what scenarios would either strategy be preferable?

Like more traditional methods, ML applications relying on simulated training data must make modeling assumptions. To what extent can ML methods be made more robust to these assumptions (e.g., by appropriately designing the feature vector, as done by S/HIC, or through simulating a greater breadth of training examples)?

ML methods have the ability to infer the values of multiple parameters simultaneously. How feasible will parameter estimation be in more complex evolutionary models using ML tools such as deep neural networks?

As described here, supervised ML relies on summaries of population genetic data as feature vectors, but what summaries are best, and can we do better than standard population genetic statistics? The recent rise of convolutional neural networks for image recognition suggests that encoding alignments as images might enable more powerful population genetics inferences – how best can we encode population genetic data?

Can we use ML to infer structured output in population genetics such as genealogies or ancestral recombination graphs?

A type of ANN called generative adversarial networks has been shown to generate data examples that can mimic true data with increasing accuracy. Can such methods be used as a substitute for population genetic simulation, perhaps to generate very large samples and chromosomes that are computationally costly to simulate?

Applications of supervised ML to population genetic data can be relatively involved, necessitating simulating data, encoding both simulated and real data as feature vectors, training the algorithm, and applying it. Can efforts to create self-contained, efficient, and user-friendly software packages capable of performing this entire workflow streamline this approach and make it more accessible to researchers?

While point estimation of population genetic model parameters is important, equally important is establishing credible intervals on our parameter estimates. How can we most effectively use ML for estimating intervals associated with parameter estimates?

Acknowledgments

We thank Alexander Xue, Matt Hahn, Parul Johri, Peter Ralph, Jeff Ross-Ibarra, Michael Blum, Adam Siepel, the laboratory of Yun Song, and one anonymous reviewer for comments on this manuscript. We also thank Justin Blumenstiel and Lex Flagel for discussions about image classification in population genetics. D.R.S. was supported by National Institutes of Health (NIH) award K99HG008696. A.D.K. was supported by NIH award R01GM117241.

Glossary

Artificial neural network (ANN)
a network of layers of one or more ‘neurons’ which receive inputs from each neuron in the previous layer, and perform a linear combination on these inputs which is then passed through an activation function. The first layer is the input layer (i.e., the feature vector) and the last layer is the output layer yielding the predicted responses. Intervening layers are referred to as ‘hidden’ layers.

Binary classification
a classification task in which there are two possible class labels, often termed positives and negatives.

Boosting
a class of machine learning (ML) techniques that seek to iteratively construct a set of predictors, weighing the influence of each predictor on the final prediction according to its individual accuracy. In addition, in most algorithms the new predictor to be added to the set focuses on examples that the current set of predictors has struggled with.

Classification
an ML task where the value to be predicted for each example is a categorical label.

Confusion matrix
a table for visualizing accuracy in multi-class classification, which is simply the contingency table of the true and predicted classes for each example in a test set (Figure 2Figure II in Box 2 for an example).

Decision tree
a hierarchical structure that predicts the response variable of an example by examining a feature, and branching to the right subtree if the value of that feature is greater than some threshold, and branching to the left otherwise. At the next level of the tree another feature is examined. The predicted value is determined by which leaf of the tree is reached at the end of this process.

Deep learning
learning using ANNs or similarly networked algorithmic models that contain multiple ‘hidden’ layers between the input and output layers.

Feature vector

a multidimensional representation of a datapoint made up of measurements (or features) taken from it (e.g., a vector of population genetic summary statistics measured in a genomic region).

1000 Genomes Dataset
a consortium project seeking to describe the breadth of human genetic variation. The 1000 Genomes Project ran from 2008 to 2015 and eventually consisted of 2504 individual genome sequences from 26 populations.

Hard sweep
a selective sweep from a *de novo* beneficial mutation. Hard sweeps are associated with large perturbations in patterns of linked genetic variation.

Labeled data
data examples for which the true response value (or label) is known.

Loss function
a measure of how correctly the response variable of an example was predicted.

***N*-fold cross-validation**
when only a small set of labeled data are available, cross-validation can be used to measure accuracy. This process partitions the labeled data into n non-overlapping equally sized sets, and trains the predictor on the union of $n-1$ of these before testing on the remaining set. This is repeated n times such that each of the n sets is used as the test set exactly once, and the average accuracy is recorded.

Overfitting
when a model has achieved excellent accuracy on a training dataset but does not generalize well – in other words the model has been tuned to precisely recognize the patterns of noise in this set that are unlikely to be present in an independent test set. Sometimes referred to as overtraining.

Precision
in binary classification, the fraction of all examples classified as positives that are true positives (i. e., the number of true positives divided by the sum of the number of true positives and number of false positives). Also known as the positive predictive value.

Random forest
an ensemble of semi-randomly generated decision trees. An example is run through each tree in the forest, and these trees then vote to determine the predicted value. Random forests can perform both classification and regression.

Recall
in binary classification, the fraction of all positives that are correctly predicted as such (i.e., the number of true positives divided by the sum of the number of true positives and number of false negatives). Also known as sensitivity.

Regression

an ML task where the value to be predicted for each example is a continuous number.

Risk function
a measure of aggregated loss across an entire training set (e.g., the expected value of the loss function). We wish the minimize the value of the risk function during training.

Site frequency spectrum (SFS)
the distribution of allele frequencies in a population sample.

Soft sweep
a selective sweep from a standing variant. In this model a mutation arises that was neutral, or nearly so, and thus drifts in a population until such a time that the environment changes and the mutation becomes selectively favored.

Support vector machine (SVM)
an ML approach that seeks to find the hyperplane that optimally separates two classes of training data. These data are often mapped to high-dimensional space using a kernel function. Variations of this approach can be performed to accomplish multiclass classification or regression.

Test set
a set of labeled examples for use during testing that is independent of the training set.

Training
the process of algorithmically generating from a training set a function that seeks to correctly predict the response variable of a datum by examining its feature vector.

Training set
a set of labeled examples for use during training.

References

1. Breiman L. Statistical modeling: the two cultures (with comments and a rejoinder by the author). Stat Sci. 2001; 16:199–231.
2. Elyashiv E, et al. A genomic map of the effects of linked selection in Drosophila. PLoS Genet. 2016; 12:e1006130. [PubMed: 27536991]
3. Hinton G, et al. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. IEEE Signal Process Mag. 2012; 29:82–97.
4. Sebastiani F. Machine learning in automated text categorization. ACM Comput Surv. 2002; 34:1–47.
5. Krizhevsky, A., et al. Imagenet classification with deep convolutional neural networks. In: Fereira, F., editor. Advances in Neural Information Processing Systems 25. Neural Information Processing Systems Foundation; 2012. p. 1097-1105.
6. Angermueller C, et al. Deep learning for computational biology. Mol Syst Biol. 2016; 12:878. [PubMed: 27474269]
7. Byvatov E, Schneider G. Support vector machine applications in bioinformatics. Appl Bioinform. 2003; 2:67.
8. Libbrecht MW, Noble WS. Machine learning applications in genetics and genomics. Nat Rev Genet. 2015; 16:321–332. [PubMed: 25948244]
9. Lin K, et al. Distinguishing positive selection from neutral evolution: boosting the performance of summary statistics. Genetics. 2011; 187:229–244. [PubMed: 21041556]

10. Mailund T, et al. Estimating divergence time and ancestral effective population size of Bornean and Sumatran orangutan subspecies using a coalescent hidden Markov model. PLoS Genet. 2011; 7:e1001319. [PubMed: 21408205]
11. Kern AD, Haussler D. A population genetic hidden Markov model for detecting genomic regions under selection. Mol Biol Evol. 2010; 27:1673–1685. [PubMed: 20185453]
12. Boitard S, et al. Detecting selective sweeps: a new approach based on hidden Markov models. Genetics. 2009; 181:1567–1578. [PubMed: 19204373]
13. Pavlidis P, et al. Searching for footprints of positive selection in whole-genome SNP data from nonequilibrium populations. Genetics. 2010; 185:907–922. [PubMed: 20407129]
14. Pybus M, et al. Hierarchical boosting: a machine-learning framework to detect and classify hard selective sweeps in human populations. Bioinformatics. 2015; 31:3946–3952. [PubMed: 26315912]
15. Ronen R, et al. Learning natural selection from the site frequency spectrum. Genetics. 2013; 195:181–193. [PubMed: 23770700]
16. Schrider DR, Kern AD. Inferring selective constraint from population genomic data suggests recent regulatory turnover in the human brain. Genome Biol Evol. 2015; 7:3511–3528. [PubMed: 26590212]
17. Schrider DR, Kern AD. S/HIC: robust Identification of soft and hard sweeps using machine learning. PLoS Genet. 2016; 12:e1005928. [PubMed: 26977894]
18. Sheehan S, Song YS. Deep learning for population genetic inference. PLoS Comput Biol. 2016; 12:e1004845. [PubMed: 27018908]
19. Amit Y, Geman D. Shape quantization and recognition with randomized trees. Neural Comput. 1997; 9:1545–1588.
20. Chen, D., et al. Blessing of dimensionality: high-dimensional feature and its efficient compression for face verification. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition; IEEE; 2013. p. 3025-3032.
21. Kotsiantis SB, et al. Supervised machine learning: a review of classification techniques. Artif Intell Rev. 2006; 26:159–190.
22. Ghahramani, Z., et al. Unsupervised learning. In: Bousquet, O., editor. Advanced Lectures on Machine Learning. Springer; 2004. p. 72-112.
23. Novembre J, et al. Genes mirror geography within Europe. Nature. 2008; 456:98. [PubMed: 18758442]
24. Rabiner LR. A tutorial on hidden Markov models and selected applications in speech recognition. Proc IEEE. 1989; 77:257–286.
25. Felsenstein J, Churchill GA. A hidden Markov model approach to variation among sites in rate of evolution. Mol Biol Evol. 1996; 13:93–104. [PubMed: 8583911]
26. Siepel A, et al. Evolutionarily conserved elements in vertebrate, insect, worm, and yeast genomes. Genome Res. 2005; 15:1034–1050. [PubMed: 16024819]
27. Dutheil JY, et al. Ancestral population genomics: the coalescent hidden Markov model approach. Genetics. 2009; 183:259–274. [PubMed: 19581452]
28. Hobolth A, et al. Genomic relationships and speciation times of human, chimpanzee, and gorilla inferred from a coalescent hidden Markov model. PLoS Genet. 2007; 3:e7. [PubMed: 17319744]
29. Boitard S, et al. Detecting selective sweeps from pooled next-generation sequencing samples. Mol Biol Evol. 2012; 29:2177–2186. [PubMed: 22411855]
30. Liang, P., Jordan, MI. An asymptotic analysis of generative, discriminative, and pseudolikelihood estimators. Proceedings of the 25th International Conference on Machine Learning; ACM; 2008. p. 584-591.
31. Anderson J, et al. The more, the merrier: the blessing of dimensionality for learning large Gaussian mixtures. Proc Mach Learn Res. 2014; 35:1135–1164.
32. Bengio Y, LeCun Y. Scaling learning algorithms towards AI. Large Scale Kernel Mach. 2007; 34:1–41.
33. Poggio T, et al. Why and when can deep-but not shallow-networks avoid the curse of dimensionality: a review. Int J Autom Comput. 2017; 14:503–519.

34. Fay JC, Wu CI. Hitchhiking under positive Darwinian selection. Genetics. 2000; 155:1405–1413. [PubMed: 10880498]
35. Fu YX, Li WH. Statistical tests of neutrality of mutations. Genetics. 1993; 133:693–709. [PubMed: 8454210]
36. Fu YX. Statistical tests of neutrality of mutations against population growth, hitchhiking and background selection. Genetics. 1997; 147:915–925. [PubMed: 9335623]
37. Kelly JK. A test of neutrality based on interlocus associations. Genetics. 1997; 146:1197–1206. [PubMed: 9215920]
38. Kim Y, Nielsen R. Linkage disequilibrium as a signature of selective sweeps. Genetics. 2004; 167:1513–1524. [PubMed: 15280259]
39. Tajima F. Statistical method for testing the neutral mutation hypothesis by DNA polymorphism. Genetics. 1989; 123:585–595. [PubMed: 2513255]
40. Voight BF, et al. A map of recent positive selection in the human genome. PLoS Biol. 2006; 4:e72. [PubMed: 16494531]
41. Bengio Y. Learning deep architectures for AI. Found Trends Mach Learn. 2009; 2:1–127.
42. Beaumont MA, et al. Approximate Bayesian computation in population genetics. Genetics. 2002; 162:2025–2035. [PubMed: 12524368]
43. Dunham I, et al. An integrated encyclopedia of DNA elements in the human genome. Nature. 2012; 489:57–74. [PubMed: 22955616]
44. Altshuler DM, et al. An integrated map of genetic variation from 1,092 human genomes. Nature. 2012; 491:56–65. [PubMed: 23128226]
45. Simonsen KL, et al. Properties of statistical tests of neutrality for DNA polymorphism data. Genetics. 1995; 141:413–429. [PubMed: 8536987]
46. Meyer KA, et al. Differential gene expression in the human brain is associated with conserved, but not accelerated, noncoding sequences. Mol Biol Evol. 2017; 34:1217–1229. [PubMed: 28204568]
47. Pollard KS, et al. Forces shaping the fastest evolving regions in the human genome. PLoS Genet. 2006; 2:e168. [PubMed: 17040131]
48. Maynard Smith J, Haigh J. The hitch-hiking effect of a favourable gene. Genet Res. 1974; 23:23–35. [PubMed: 4407212]
49. Nielsen R, et al. A scan for positively selected genes in the genomes of humans and chimpanzees. PLoS Biol. 2005; 3:e170. [PubMed: 15869325]
50. Breiman L. Random forests. Mach Learn. 2001; 45:5–32.
51. Geurts P, et al. Extremely randomized trees. Mach Learn. 2006; 63:3–42.
52. Hermisson J, Pennings PS. Soft sweeps molecular population genetics of adaptation from standing genetic variation. Genetics. 2005; 169:2335–2352. [PubMed: 15716498]
53. Orr HA, Betancourt AJ. Haldane's sieve and adaptation from the standing genetic variation. Genetics. 2001; 157:875–884. [PubMed: 11157004]
54. Gutenkunst RN, et al. Inferring the joint demographic history of multiple populations from multidimensional SNP frequency data. PLoS Genet. 2009; 5:e1000695. [PubMed: 19851460]
55. Hey J, Nielsen R. Integration within the Felsenstein equation for improved Markov chain Monte Carlo methods in population genetics. Proc Natl Acad Sci. 2007; 104:2785–2790. [PubMed: 17301231]
56. Li H, Durbin R. Inference of human population history from individual whole-genome sequences. Nature. 2011; 475:493–496. [PubMed: 21753753]
57. Liu X, Fu YX. Exploring population size changes using SNP frequency spectra. Nat Genet. 2015; 47:555–559. [PubMed: 25848749]
58. Sheehan S, et al. Estimating variable effective population sizes from multiple genomes: a sequentially Markov conditional sampling distribution approach. Genetics. 2013; 194:647–662. [PubMed: 23608192]
59. Beaumont MA. Approximate Bayesian computation in evolution and ecology. Annu Rev Ecol Evol Syst. 2010; 41:379–406.
60. Aeschbacher S, et al. A novel approach for choosing summary statistics in approximate Bayesian computation. Genetics. 2012; 192:1027–1047. [PubMed: 22960215]

61. Jiang, B., et al. Learning summary statistic for approximate Bayesian computation via deep neural network. arXiv. 2015. https://arxiv.org/abs/1510.02175
62. Blum MG, François O. Non-linear regression models for approximate Bayesian computation. Stat Comput. 2010; 20:63–73.
63. Pudlo P, et al. Reliable ABC model choice via random forests. Bioinformatics. 2016; 32:859–866. [PubMed: 26589278]
64. Schrider, D., et al. Supervised machine learning reveals introgressed loci in the genomes of *Drosophila simulans* and *D. sechellia*. bioRxiv. 2017. https://www.biorxiv.org/content/early/2017/07/31/170670
65. Adrian AB, et al. Predictive models of recombination rate variation across the *Drosophila melanogaster* genome. Genome Biol Evol. 2016; 8:2597–2612. [PubMed: 27492232]
66. Gao F, et al. New software for the fast estimation of population recombination rates (FastEPRR) in the genomic era. Genes Genomes Genet. 2016; 6:1563–1571.
67. Lin K, et al. A fast estimate for the population recombination rate based on regression. Genetics. 2013; 194:473–484. [PubMed: 23589457]
68. McVean GA, et al. The fine-scale structure of recombination rate variation in the human genome. Science. 2004; 304:581–584. [PubMed: 15105499]
69. Ewing GB, Jensen JD. The consequences of not accounting for background selection in demographic inference. Mol Ecol. 2016; 25:135–141. [PubMed: 26394805]
70. Schrider DR, et al. Effects of linked selective sweeps on demographic inference and model selection. Genetics. 2016; 204:1207–1223. [PubMed: 27605051]
71. Sermanet, P., et al. Overfeat: Integrated recognition, localization and detection using convolutional networks. arXiv. 2013. https://arxiv.org/abs/1312.6229
72. Graham, B. Fractional max-pooling. arXiv. 2014. https://arxiv.org/abs/1412.6071
73. Yu, C-NJ., Joachims, T. Learning structural SVMs with latent variables. Proceedings of the 26th Annual International Conference on Machine Learning; ACM; 2009. p. 1169-1176.
74. Watterson G. On the number of segregating sites in genetical models without recombination. Theor Popul Biol. 1975; 7:256–276. [PubMed: 1145509]
75. Hudson RR. Generating samples under a Wright–Fisher neutral model of genetic variation. Bioinformatics. 2002; 18:337–338. [PubMed: 11847089]
76. Pedregosa F, et al. Scikit-learn: machine learning in Python. J Mach Learn Res. 2011; 12:2825–2830.
77. Nei M, Li WH. Mathematical model for studying genetic variation in terms of restriction endonucleases. Proc Natl Acad Sci. 1979; 76:5269–5273. [PubMed: 291943]
78. Garud NR, et al. Recent selective sweeps in North American *Drosophila melanogaster* show signatures of soft sweeps. PLoS Genet. 2015; 11:e1005004. [PubMed: 25706129]
79. Messer PW, Petrov DA. Population genomics of rapid adaptation by soft selective sweeps. Trends Ecol Evol. 2013; 28:659–669. [PubMed: 24075201]
80. Cortes C, Vapnik V. Support-vector networks. Mach Learn. 1995; 20:273–297.
81. Quinlan JR. Induction of decision trees. Mach Learn. 1986; 1:81–106.
82. Schapire RE. The strength of weak learnability. Mach Learn. 1990; 5:197–227.
83. Bishop, CM. Neural Networks for Pattern Recognition. Oxford University Press; 1995.
84. LeCun Y, et al. Deep learning. Nature. 2015; 521:436–444. [PubMed: 26017442]
85. Schrider DR, et al. Soft shoulders ahead: spurious signatures of soft and partial selective sweeps result from linked hard sweeps. Genetics. 2015; 200:267–284. [PubMed: 25716978]
86. Jensen JD, et al. Distinguishing between selective sweeps and demography using DNA polymorphism data. Genetics. 2005; 170:1401–1410. [PubMed: 15911584]
87. Nielsen R, et al. Genomic scans for selective sweeps using SNP data. Genome Res. 2005; 15:1566–1575. [PubMed: 16251466]
88. Schrider DR, Kern AD. Soft sweeps are the dominant mode of adaptation in the human genome. Mol Biol Evol. 2017; 34:1863–1877. [PubMed: 28482049]

89. Blum MG, et al. A comparative review of dimension reduction methods in approximate Bayesian computation. Stat Sci. 2013; 28:189–208.

90. Marin, J-M., et al. ABC random forests for Bayesian parameter inference. arXiv. 2016. https://arxiv.org/abs/1605.05537

Highlights

ML methods are powerful approaches that have revolutionized many fields, but their use in population genetics inference is only beginning.

These methods are able to take advantage of high dimensional input – an important asset for population genetics inference – and are often more robust than other statistical approaches.

The early applications of ML to population genetics demonstrate that they outperform traditional approaches.

In this review we introduce ML to a biology audience, discuss examples of their application to evolutionary and population genetics, and lay out future directions that we view as promising.

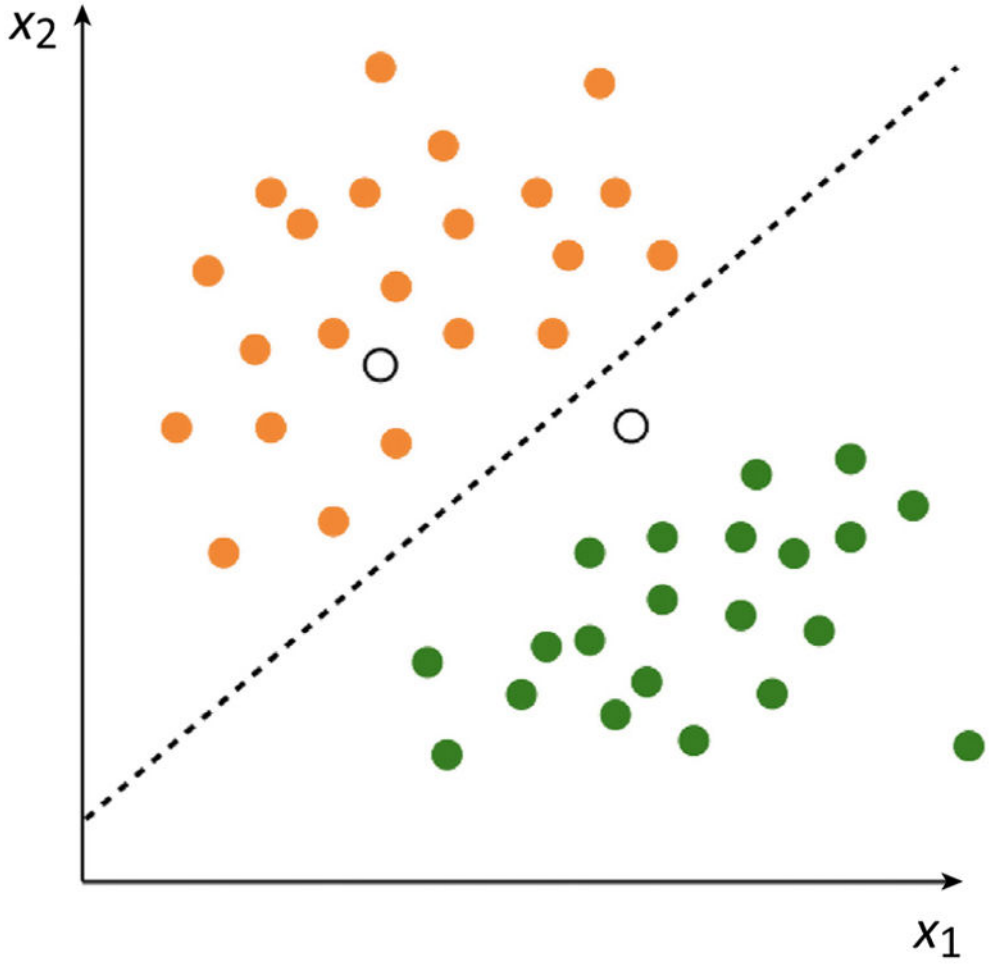

Figure I. An Imaginary Training Set of Two Types of Fruit, Oranges (Orange Filled Points) and Apples (Green Filled Points), Where Two Measurements Were Made for Each Fruit

With a training set in hand we can use supervised ML to learn a function that can differentiate between classes (broken line) such that the unknown class of new datapoints (unlabeled points above) can be predicted.

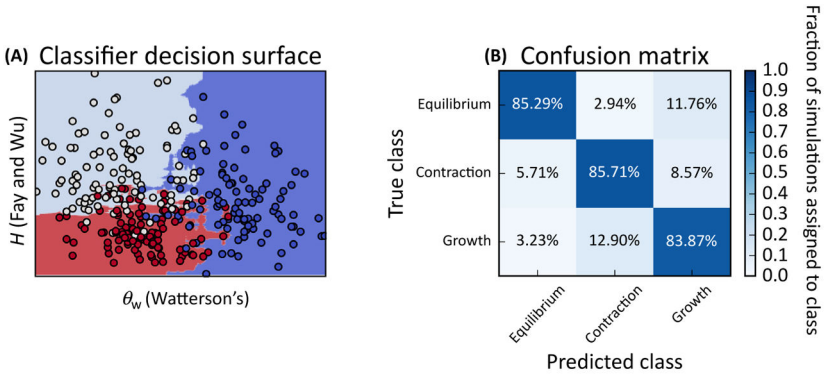

Figure II. An Example Application of Supervised ML to Demographic Model Selection

In this example population samples experiencing constant population size (equilibrium), a recent instantaneous population decline (contraction), or recent instantaneous expansion (growth) were simulated. A variant of a random forest classifier [51] was trained, which is an ensemble of semi-randomly generated decision trees, to discriminate between these three models on the basis of a feature vector consisting of two population genetic summary statistics [34,74]. (A) The decision surface: red points represent the growth scenario, dark-blue points represent equilibrium, and light-blue points represent contraction. The shaded areas in the background show how additional datapoints would be classified – note the non-linear decision surface separating these three classes. (B) The confusion matrix obtained from measuring classification accuracy on an independent test set. Data were simulated using ms [75], and classification was performed via scikitlearn [76]. All code used to create these figures can be found in a collection of Jupyter notebooks that demonstrate some simple examples of using supervised ML for population genetic inference provided here: https://github.com/kern-lab/popGenMachineLearningExamples.

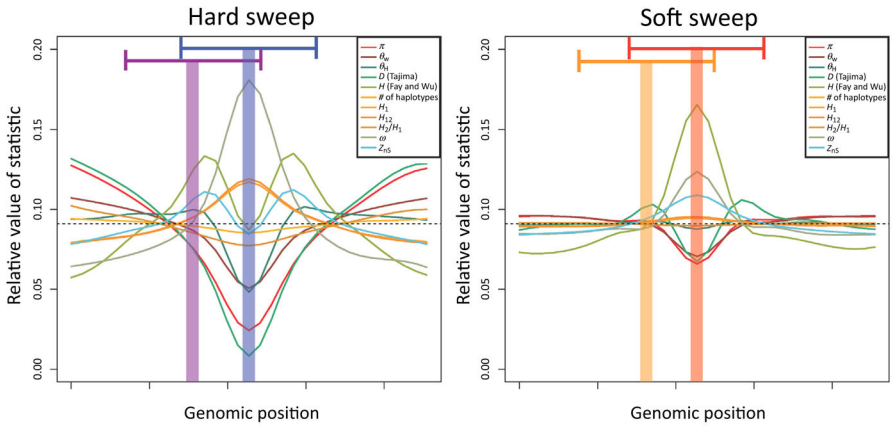

Figure III. A Visualization of S/HIC Feature Vector and Classes

The S/HIC feature vector consists of π [77], $\widehat{\theta_w}$ [74], $\widehat{\theta_H}$ [34], the number (#) of distinct haplotypes, average haplotype homozygosity, H_{12} and H_2/H_1 [78,79], Z_{nS} [37], and the maximum value of ω [48]. The expected values of these statistics are shown for genomic regions containing hard and soft sweeps (as estimated from simulated data). Fay and Wu's H [34] and Tajima's D [39] are also shown, though these may be omitted from the vector because they are redundant with π, $\widehat{\theta_w}$, and $\widehat{\theta_H}$. To classify a given region the spatial patterns of these statistics are examined across a genomic window to infer whether the center of the window contains a hard selective sweep (blue shaded area on the left, using statistics calculated within the larger blue window), is linked to a hard sweep (purple shaded area and larger window, left), contains a soft sweep (red, on the right), is linked to soft sweep (orange, right), or is evolving neutrally (not shown).