The Unreasonable Effectiveness of Convolutional Neural Networks in Population Genetic Inference

Lex Flagel[1,2], Yaniv Brandvain[2], and Daniel R. Schrider[3,*]

[1]Monsanto Company, Chesterfield, MO.

[2]Department of Plant and Microbial Biology, University of Minnesota, St. Paul, MN.

[3]Department of Genetics, University of North Carolina, Chapel Hill, NC.

*Corresponding author. Email: drs@unc.edu

ABSTRACT

Population-scale genomic datasets have given researchers incredible amounts of information from which to infer evolutionary histories. Concomitant with this flood of data, theoretical and methodological advances have sought to extract information from genomic sequences to infer demographic events such as population size changes and gene flow among closely related populations/species, construct recombination maps, and uncover loci underlying recent adaptation. To date most methods make use of only one or a few summaries of the input sequences and therefore ignore potentially useful complementary information encoded in the data. The most sophisticated of these approaches involve likelihood calculations, which require theoretical advances for each new problem addressed, and often must focus on a single aspect of the data (e.g. allele frequency information only) in the interest of mathematical and computational tractability. Directly interrogating the entirety of the input sequence data in a likelihood-free manner would thus offer a fruitful alternative. Here we accomplish this by representing DNA sequence alignments as images and then using a deep learning technique called a convolutional neural network (CNN) to make population genetic inferences from these images. We apply CNNs to a number of evolutionary questions and find that they frequently exceed current state-of-the-art methods. Importantly, we show that CNNs can perform accurate evolutionary model selection and parameter estimation, even on problems that have not received detailed theoretical treatments. Thus, when applied to population genetic alignments, CNN are capable of outperforming expert-derived statistical methods, and offer a new path forward in cases where no theoretical approaches exist.

INTRODUCTION

Using genetic data to make inferences about the natural histories of populations represents a major goal of evolutionary research. As the ever-increasing throughput of DNA sequencing technologies makes the generation of large population genomic data sets more routine, researchers can leverage patterns of genetic variation across the genome to characterize the evolutionary forces at play (Hahn 2008). For example, advances have been made in identifying historical demographic events such as population size changes (Marth *et al.* 2004; Tennessen *et al.* 2012; Gazave *et al.* 2014) and genetic exchange between populations and species (Martin *et al.* 2013; Hellenthal *et al.* 2014; Sankararaman *et al.* 2014; Corbett-Detig and Nielsen 2017; Schrider *et al.* 2018). Population genomic analyses have also revealed the pervasive impact of selection on polymorphism on linked neutral polymorphism (Begun and Aquadro 1992; Begun *et al.* 2007; Langley *et al.* 2012; Elyashiv *et al.* 2016), both through positive selection (Maynard Smith and Haigh 1974; Kaplan *et al.* 1989) and purifying selection (Charlesworth *et al.* 1993). As the volume of population genomic data sets has increased, so too has the demand for powerful computational methods capable of using these data to learn about the fundamental evolutionary processes shaping genomic variation.

To meet this need, myriad statistical and computational tools have been devised to answer evolutionary questions using population genetic data. One particularly common paradigm, which predates the high-throughput sequencing revolution, is that of the population genetic summary statistic: a single value designed to capture the information present in a sequence alignment of individuals from one or more populations. When a particular evolutionary phenomenon acts on a population, it alters the shapes of genealogies, and this effect is then manifest in the observed sequence alignment. For example, a population expansion will result in genealogies with longer branches near the leaves of the tree, which will manifest as an excess of rare alleles. Many summary statistics seek to uncover the signature of these genealogical skews through their effect on the alignment; e.g. Tajima's D will be negative following a recent expansion or recovery from a bottleneck (Tajima 1989; Simonsen *et al.* 1995). Ideally a summary statistic will only detect the signal of the evolutionary process it is being used to investigate, but in practice summary statistics are frequently confounded by other forces that may have similar effects on the shapes and/or sizes of genealogies. For example, Tajima's D is sensitive to positive selection as well as population size changes (Simonsen *et al.* 1995).

Rather than detecting a phenomenon of interest, some statistics are designed to estimate theoretical population parameters such as the population-scaled mutation rate $\theta=4Nu$ (Watterson 1975; Nei and Li 1979; Tajima 1989; Fay and Wu 2000; Achaz 2009) or recombination rate $\rho=4Nr$ (Hudson and Kaplan 1985; Hudson 1987). Unfortunately, these statistics will also be sensitive to confounding forces, and will only estimate the parameter of interest when none of its assumptions are violated. Moreover, neither type of summary statistic will capture all of the information present in the alignment. Thus a major challenge of population genetic inference is to create methods that utilize as much information from the input data as possible in order to maximize our ability to distinguish among the numerous evolutionary processes that can give rise to an observed signal.

One approach researchers have adopted to address this challenge is to incorporate a larger number of observations from the data into likelihood-based inference methods. However, calculating likelihoods of population genomic data sets is often mathematically and computationally intractable, and therefore such approaches often ignore the non-independence of observations (e.g. using composite likelihoods; Hudson 2001; Nielsen *et al.* 2005). For example, Nielsen et al.'s SweepFinder (2005), which examines allele frequencies at polymorphisms flanking a focal region to determine whether that region has experienced a recent selective sweep (Maynard Smith and Haigh 1974), treats each allele frequency as an independent observation despite the partially shared evolutionary histories linked alleles experience. Another drawback of most likelihood-based methods is that they generally compute the likelihood of very few features of the data (often only one), and therefore additional information that could improve accuracy is ignored. For example, SweepFinder examines allele frequencies but ignores linkage disequilibrium (LD), which is elevated in areas flanking the selected site (Kim and Nielsen 2004).

More recently, population geneticists have begun to explore an alternative strategy of using a large set of complementary summary statistics for model selection and parameter estimation, an approach that often results in more powerful and robust inference (e.g. Lin *et al.* 2011; Pybus *et al.* 2015; Gao *et al.* 2016; Schrider and Kern 2016; Sheehan and Song 2016). Each summary statistic seeks to measure a particular attribute of the genealogy, and one can thus design a customized set of summary statistics to more fully represent the genealogical information present in the sequence alignment. This view deploys summary statistics less for their individual links to underlying theory, and more for their collective ability to perform pattern recognition.

The challenge then becomes extracting information about the underlying evolutionary processes from the set of summary statistics. Two exciting approaches for dealing with this challenge that have garnered increasing attention in recent years are approximate Bayesian computation (ABC; reviewed in Beaumont 2010) and supervised machine learning (reviewed in Schrider and Kern 2018). Both of these approaches make use of suites of user-defined summary statistics and training data generated under known parameters to identify reasonable evolutionary models and parameterizations that could have generated the observed data. Here we focus on the supervised machine learning approach, as it sets the scene for the convolutional neural networks described below.

In the terminology of supervised machine learning, each summary statistic is called a feature, and the full set of statistics used is called a feature vector. To use supervised machine learning, a researcher must first obtain training data (often referred to as "labeled" data)—a set of data points for each of which the user has obtained a feature vector (the explanatory variables) accompanied by a known outcome (the response variable). Next, a supervised machine learning algorithm is trained to predict the outcome given the feature vector using the labeled training data. Thus, the supervised machine learning technique automates the process of extracting information and constructing rules from a set of summary statistics. Across many areas of research, supervised machine learning techniques are fast replacing rules developed by human experts because they are often more accurate (LeCun *et al.* 2015).

Machine learning methods are increasingly applied to numerous problems in population genomics (Schrider and Kern 2018). This approach requires the generation of labeled training data via population genetic simulation, an endeavor that has grown considerably more feasible given recent improvements in simulation flexibility and efficiency (e.g. Thornton 2014; Kelleher *et al.* 2016; Haller and Messer 2017; Kelleher *et al.* 2018). To date, population genetic applications of machine learning include demographic inference (Pudlo *et al.* 2016; Sheehan and Song 2016), local ancestry inference (Schrider *et al.* 2018), inferring recombination rates (Lin *et al.* 2013; Gao *et al.* 2016), and detecting genomic regions experiencing recent selective sweeps (Pavlidis *et al.* 2010; Lin *et al.* 2011; Ronen *et al.* 2013; Pybus *et al.* 2015; Schrider and Kern 2016). While such methods have great promise, they still rely on a user-defined set of summary statistics. Moreover, it is not known whether it is possible to construct a set of statistics that sufficiently captures all relevant information in the input data.

Unlike other machine learning approaches, convolutional neural networks (CNN; LeCun *et al.* 1998) are pattern recognition algorithms that do not require a predefined feature vector. When fed labeled training data, a CNN discovers meaningful features itself, in essence making a feature vector, and then extracts information from these features in order to make inferences. CNNs have proved effective in a number of fields (reviewed in LeCun *et al.* 2015), and particularly in the field of image recognition, where they have achieved dramatic improvements over previous efforts (e.g. Lawrence *et al.* 1997; Krizhevsky *et al.* 2012; Simonyan and Zisserman 2014). The application of CNNs to population genomic inference is just beginning, and shows great promise (Chan *et al.* 2018; Kern and Schrider 2018). Population genetic questions may be particularly well suited for CNN-based learning because they take matrices as inputs, and alignments of sequenced chromosomes are quite naturally represented in this manner.

The goal of this paper is to assess the effectiveness of CNNs as a general strategy for population genomic inference. We demonstrate that CNNs can be successfully applied to a number of population genomic problems, in some cases achieving surprising accuracy. In particular, we use simulation to show that CNNs can leverage images of aligned sequences to accurately detect selective sweeps, uncover regions experiencing gene flow between related populations/species, make demographic inferences, and estimate recombination rates. Indeed, in most cases we observe performance that matches or exceeds that of current state-of-the-art methods. We also use a CNN to accurately infer recombination rates from read coverage data in a simulated autotetraploid, demonstrating this approach's flexibility in handling noisy data while solving a complex problem for which no theoretical solutions exist. In light of these encouraging findings, we argue that population genetics researchers should consider CNNs as a potential solution to a variety of problems involving evolutionary inferences from sequence data. Because many readers will have little background with this tool, we also provide an overview of the inner workings of CNNs and explore several technical considerations that may impact performance.

MATERIALS AND METHODS

Overview of convolutional neural networks

Internally, a CNN is a type of artificial neural network – a collection of connected layers of combinatorially linked logic functions (termed *artificial neurons*) that take an input and transform it into an output value (Mitchell 1997). In a typical fully connected artificial neural network, the

input values are fed through a series of layers of neurons (Figure 1A), termed hidden layers, before reaching the output layer which transforms its inputs into a final prediction. The output for the jth neuron within one of the hidden layers is given by the following:

$$f\left(\sum_{i}^{n} w_{ij}x_i + b_j\right)$$

In the expression above, x_i is the neuron's i^{th} input value (either an input value from the data or from a neuron in the previous layer's output), and w_{ij} is the *weight* attached to the connection between that node (i) and the current node (j) and b_j is the current node's *bias* term. That is, to obtain the value of neuron j, we compute the linear combination of the vector containing all values from the previous layer and the jth neuron's vector of weights; the results of this summation are in turn added to neuron j's bias term and then fed as input to some function f, termed the *activation function*. Thus, an artificial neural network is a mathematical function.

Importantly, by changing the values of the weights and biases, an artificial neural network can be tuned to detect informative patterns in the input data in order to produce the desired output. In the case of image recognition, an image is first represented numerically, typically as a matrix of pixel intensities, and then transformed by the artificial neural network to produce an output, for example a prediction of the type of object in the image. CNNs (Figure 1B–C) differ from standard artificial neural networks in that they begin with one or more convolutional layers, in which a series of smaller weight matrices referred to as "filters" slide across the input image—mimicking the manner in which animal cortical neurons each focus on input only from a small receptive field—and perform a matrix convolution at each step until a series of filtered image matrices are produced (LeCun *et al.* 1998). Each convolutional layer is often followed by a pooling layer which reduces the size of these filtered image matrices while maintaining potentially important discriminatory information obtained by the convolutional filters. Finally, these matrices are flattened and fed into a fully connected (or "dense") artificial neural network (for an accessible overview see LeCun *et al.* 2015). Thus, salient features derived from the image matrix by the convolutional and pooling layers are passed into one or more layers of a fully connected neural network whose output layer then yields our predicted response value.

CNNs allow for two types of convolutional layers: 1-dimensional and 2-dimensional layers, which differ only with respect to the possible shapes that the convolutional filter can take (Figure 1B–C). 1-dimensional (1D) convolutions are often used in the application to time-series

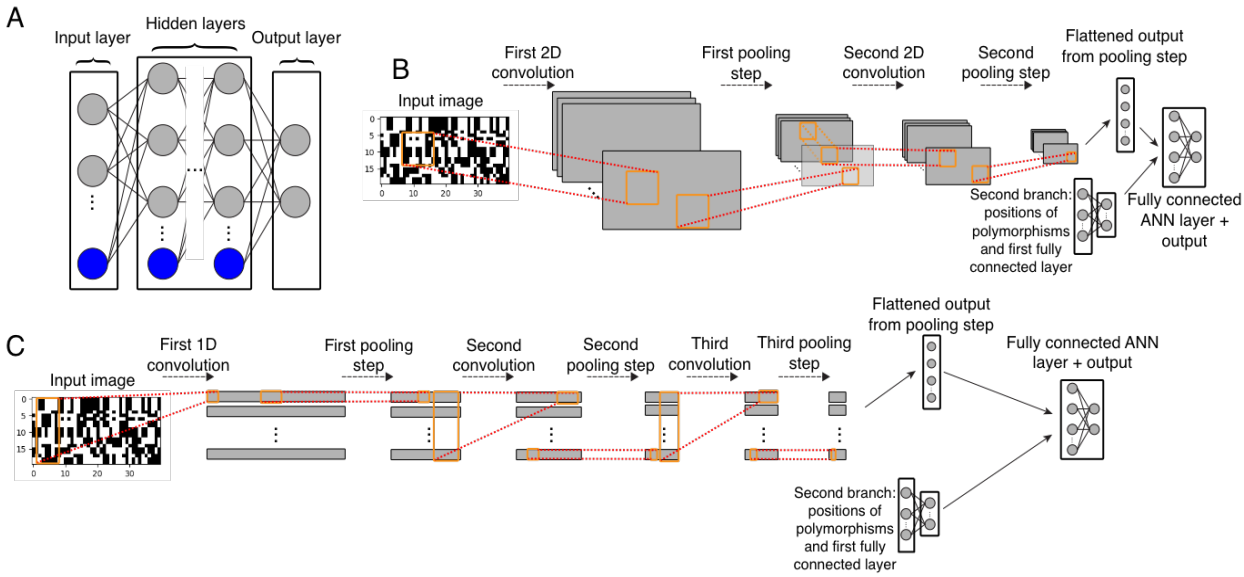

Figure 1: Schematics of a standard feedforward neural network and two convolutional neural network designs used in this study. A) Diagram of a fully connected feedforward neural network. Gray circles represent input (left side), output (right side), or hidden (center) neurons. Blue circles represent collections of bias terms. With the exception of the input layer, the value of any given neuron is a linear combination of values from the previous layer plus a bias term; this sum is then passed to an activation function (not shown). Each edge represents a distinct weighted input or bias term. Outputs may represent class membership posterior probabilities or estimates of continuous variables. B) A diagram of a 2D CNN similar to that used in this study to infer demographic parameters. The input is an alignment represented as an image which is passed through a first convolutional layer in order to create a set of feature maps which are then downsized via a pooling step. These feature maps are then passed through a second convolutional filter and pooling step, and the resulting output is flattened in order to be passed as input into a fully connected feedfoward layer (bias terms not shown). Also passed into this layer is output from a second branch of this network: the vector of positions of segregating sites in the alignment which have been passed through their own fully connected layer. Finally, the last fully connected ANN layer yields the predicted output values. C) Similar to panel B, but showing a 1D CNN with three convolutional layers (each followed by a pooling step), as used for our recombination rate estimator.

data, but are also applicable to sequence alignment matrices. Despite its name, a 1D filter is not a vector but rather a rectangular matrix that spans a user-defined number of entries (called the "kernel size") in one dimension in the input data (in our case this dimension is that of the polymorphic sites in the alignment), and stretches entirely across the other dimension (in our case across all chromosomes in the sample). A 2-dimensional (2D) convolutional filter, which is more often used with image data, allows the user to specify both dimensions of the filter matrix (often using a square matrix). Whether 1- or 2-dimensional, the benefit of incorporating convolutions is

that it allows the CNN to take advantage of structural information in the input data. For example, from an image of a face, a CNN can learn to detect the repeated pattern of the eye shape and the location of both eyes relative to one another and to other features. When there is meaningful structural information such as this, CNNs tend to outperform non-convolutional neural networks.

Here our input data is an alignment of linked segregating sites with partially shared evolutionary histories. Our hope is that a CNN can discover structural information in these data in order to make evolutionary inferences—for example, locating the valley in diversity at the center of a sweep (Maynard Smith and Haigh 1974), the "shoulders" on the flanks of a sweep where linkage disequilibrium and allele frequencies are both elevated (Schrider *et al.* 2015), or even the spatial relationship between these different structures. We also note that neural networks such as CNNs can have multiple "branches" each with separate architectures and input types—in some of the cases discussed below we will incorporate an additional network branch whose input is the vector of the positions of the segregating sites (Figure 1B–C).

Like all supervised machine learning methods, a CNN must be trained on labeled training data before it can make predictions on unlabeled data (i.e. data whose response variables are unknown). Training is accomplished by tuning the weights and biases that control the behavior of its artificial neurons so that together they maximize the accuracy of the outputs on the training data. This tuning occurs over a number of iterations using the backpropagation algorithm (Rumelhart *et al.* 1986), which in modern implementations feeds a small number of training examples (a "mini-batch") through the network and then estimates the error gradient on the output vectors produced for these examples. The error gradient is then propagated in reverse through the network—a given hidden neuron's contribution to the error is proportional to the linear combination of its weight vector and the errors associated with each neuron in the next layer. The weights are then updated using one of the many flavors of stochastic gradient descent (e.g. Kingma and Ba 2014). This process repeats until each training example has been fed through the network, marking the completion of a single training iteration. Training continues for a number of these iterations (often called epochs) until a specified stopping criterion is reached (e.g. a predefined number of iterations has been performed, accuracy on the validation set has not improved relative to the previous iteration, etc.).

222 In the context of population genetics, the CNN's input could be a matrix of allelic states
223 at each polymorphic site (Figure 2). For example, an alignment of haploid individuals M, where
224 M_{ij}=0 if the ith individual has the ancestral allele at the jth segregating site in the alignment, and 1
225 if this individual has the derived allele (an input format that can easily be altered to allow for
226 multiallelic polymorphisms); we adopt this approach and variants of it below. The output can be
227 a categorical indicator (e.g. whether or not the genomic window experienced a recent selective
228 sweep) in which the problem is referred to as a classification task in machine learning
229 terminology, a quantitative value (e.g. the population recombination rate) in which case the task
230 is referred to as regression, or a vector with numerous categorical and/or quantitative values.
231 Once the CNN has been trained to produce the desired output, it can be applied to unlabeled
232 data (e.g. sequence from natural populations).
233 Because supervised machine learning relies on predictive functions tuned algorithmically
234 from training data rather than theoretical predictions, CNNs can be flexibly applied to any
235 problem for which a training set can be obtained. In a population genetics context, coalescent
236 simulations provide a versatile and computationally efficient (Hudson 2002; Teshima and Innan
237 2009; Ewing and Hermisson 2010; Kelleher *et al.* 2016; Kern and Schrider 2016) means to
238 generating training data. In this paper we relied exclusively on coalescent simulations to produce
239 training data for the CNN. However, compute-intensive forward population simulations may
240 offer greater flexibility than coalescent simulations in some situations, and recent advances are
241 making them more computationally feasible (Kelleher *et al.* 2018).
242
243 **Computational environment for training CNNs**
244 All CNNs used in this study were developed using two open source Python packages: Keras
245 (version 2.0.6; https://keras.io/) to define neural network architecture and orchestrate training
246 and testing, and TensorFlow (version 1.1.0; https://www.tensorflow.org/) as the backend (i.e.
247 TensorFlow performs the computation during training/testing). All code used for training is
248 available online (https://github.com/flag0010/pop_gen_cnn).
249
250 **CNN validation strategy**
251 For each task, we divided our set of simulated inputs into three sets: a training set, a validation
252 set, and a test set. The training set was used to optimize the weights and biases of the CNN. The

253 validation set was used during training to determine how well the CNN generalizes to unseen
254 data, and adjustments were made to the CNN to improve its performance on the validation data.
255 Finally, the test set was used to obtain a performance assessment of the final trained network.
256 Importantly, this test set was previously unseen by the CNN and therefore yields an unbiased
257 evaluation of its accuracy.
258

259 **Evaluating techniques for rescaling and reordering inputs to improved CNN**
260 **accuracy**

261 To evaluate the impact of alternative data preparation techniques (below), we developed a simple
262 CNN that estimates the locus-wide population mutation rate $\theta=4N\mu L$ where μ is the mutation
263 rate per base pair per generation and L is the physical length of the locus being examined.
264 (https://github.com/flag0010/pop_gen_cnn/tree/master/data_prep_tricks). This CNN is
265 trained using alignment images with forty chromosomes and θ drawn uniformly between 10 and
266 50 as simulated for a panmictic, constant sized population by `ms` (Hudson 2002). We trained this
267 CNN to minimize the root mean squared error (RMSE) between its prediction and the true value
268 of θ using 4,000 training matrices. Then its accuracy was scored on 1,000 test matrices that the
269 CNN was never trained on. These values were compared under the different data preparation
270 approaches described below.

271 First, the matrices output by most coalescent simulation software, including `ms`, encode
272 ancestral and derived alleles for bialleleic sites as 0 and 1, respectively, and present the matrix
273 with chromosomes as rows and sites as columns. When doing 1D convolutions, we used row-wise
274 convolutional filters (Figure 1C), i.e. those that examine each chromosome in our sample across a
275 small number of contiguous segregating sites (specified by the "kernel_size" parameter in Keras)
276 before sliding the filter forward one site (our stride length, "strides" in Keras, was always set to 1).
277 At present Keras does not allow for row-wise 1D convolutions, so we accomplished this by simply
278 transposing the alignment matrix and performing column-wise convolutions.

279 We find that sorting the chromosomes in the alignment by genetic similarity accelerates
280 learning. For example, the matrices in Figure 2 contain identical information, but chromosomes
281 in the matrix on the left are randomized, while on the right they are sorted by genetic similarity.
282 We found that the CNN trains faster when given sorted matrices like the one on the right in

283 Figure 2. We offer a fast algorithm for sorting matrices by genetic similarity (sort_min_diff
284 function:
285 https://github.com/flag0010/pop_gen_cnn/blob/master/data_prep_tricks/genet.data.matrix.p
286 rep.tricks.py).

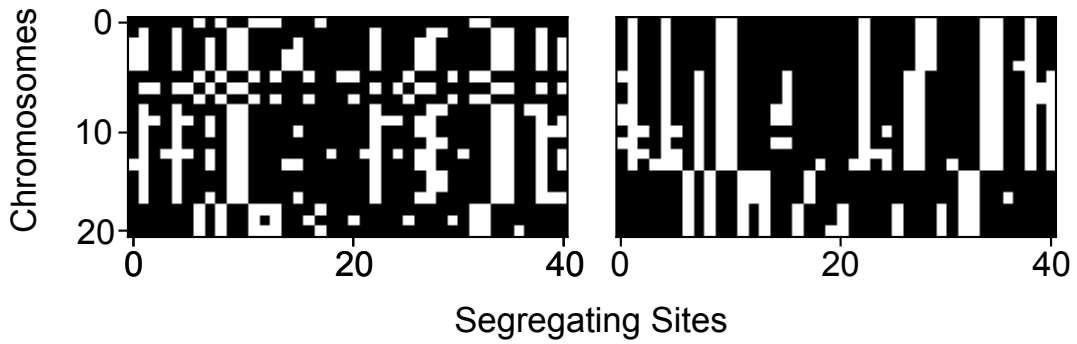

Figure 2: Example population genetic alignments visualized as black-and-white images. An unsorted alignment matrix (left) and this same matrix sorted by genetic similarity among chromosomes (right) are shown. Each row represents one of twenty chromosomes in the sample and each column represents one of forty segregating sites. Derived and ancestral states are encoded as black and white, respectively.

287

288 **Introgression detection**

289 To detect introgression, we simulated training and test examples with `msmove`
290 (https://github.com/geneva/msmove) from the same demographic model that Schrider et al.
291 (2018) used to train the FILET classifier for detecting introgression between *Drosophila simulans*
292 and *D. sechellia*. In total we produced 237,500 coalescent simulations from 3 classes: 112,500
293 without no migration between species (No Introgression), 112,500 with gene flow from *D.*
294 *simulans* into *D. sechellia* (*sim*→*sech*), and 12,500 with gene flow from *D. sechellia* into *D. simulans*
295 (*sech*→*sim*). We used fewer *sech*→*sim* examples because test runs on smaller training sets suggested
296 that the network could detect this class fairly accurately, which allowed us to increase the
297 sampling of the other two more challenging classes. The simulations were randomly assigned to
298 training and validation sets so that the training set included 107,500 examples each from the No
299 Introgression and *sim*→*sech* classes, and 7,500 examples from the *sech*→*sim* class. Both the
300 validation set and the test set contained 2,500 of each class (i.e. 7,500 total).

As in the *Drosophila* data set to which Schrider et al. applied FILET, each of our coalescent simulations generated 34 chromosomes (14 *D. sechellia* and 20 *D. simulans*). Each column in the alignment corresponded to a biallelic polymorphism, which was encoded as "0" (ancestral allele) or "1" (derived allele) for each chromosome. Each matrix was organized so that individual chromosomes were grouped by species. Each coalescent simulation produced a different number of segregating sites (with the largest containing 1201 polymorphisms). Because the CNN's input matrices must all have the same dimensions, we padded the right side of all matrices with fewer than 1201 polymorphisms with columns containing only "0" until the total number of columns reached 1201. Finally we transposed this matrix resulting in a 1201×34 matrix for each coalescent simulation.

We trained a CNN architecture with three 1D-convolutional layers (kernel size = 2), each followed by average-pooling, and finally two densely connected layers (i.e. the same network architecture as the main network branch illustrated in Figure 1C, but with one additional dense layer). These layers contained 256, 128, 128, 128, and 128 neurons, respectively. To avoid overfitting during training, each layer used dropout regularization (randomly removing 25% of neurons between convolutional layers during each training iteration, and 50% between densely connected layers) and rectified linear unit activation functions (i.e. ReLUs; Nair and Hinton 2010). This "dropout regularization" encourages the CNN to learn redundant representations of the data, thereby reducing the network's dependence on individual weights (Srivastava *et al.* 2014). The last layer was a sigmoid output layer with 3 neurons, each corresponding to the 3 classes given above. The CNN was trained using the Adam optimization procedure (Kingma and Ba 2014), a categorical cross-entropy loss function, and a mini-batch size of 256. The CNN was run for 19 training iterations through the training data.

Recombination rate: phased haplotype version

For the recombination rate estimator we used `ms` (Hudson 2002) to simulate 50 chromosomes, each with a target length of 20kb. To do so, we drew a population size (N) from the following values: 5,000, 10,000, 15,000, 20,000, and 50,000, and set the mutation rate parameter $\theta = 4N\mu L$ (letting μ=1.5x10^{-8} and again L=20kb). We also set a population-scaled recombination rate, $\rho = 4NrL$, where r is the per bp crossover rate per meiosis, by drawing r from a bounded exponential distribution ranging from10^{-8} to 10^{-6}. Following this procedure, we generated 156,275 coalescent

332 simulations. ~92% were used to train the CNN, and ~4% each were set aside for validation and
333 testing. To assess our CNNs ability to interpolate to unseen population sizes, we also created
334 5,000 additional test matrices using the procedures above, but with N drawn uniformly from the
335 following: 30,000, 35,000, 40,000, and 45,000.

336 Each simulation was represented by a matrix of 50 rows, one for each chromosome, and
337 418 columns (the largest number of segregating sites). As before we encoded the ancestral allele
338 with "0" and the derived allele with "1". Because not all simulations resulted in the same number
339 of polymorphisms, we padded both the genotype matrix and the position vector in the same
340 manner as for the introgression CNN, bringing the total size of each matrix to 50×418. Next, we
341 sorted each matrix by genetic similarity among chromosomes as described above and then
342 transposed the matrix to 418×50. We also extracted the segregating site positions vector from the
343 `ms` output which represents each position as a real number between zero (the leftmost position
344 on the simulated chromosome) and one (the rightmost position). For simulations with fewer than
345 418 segregating sites, we padded the positions vector with "-1"s.

346 Prior to training, we also transformed the ρ values for the training, validation, and test sets
347 by taking the natural log of each value and centering them on the mean of the training set. By
348 using the mean from the training set for each of these we ensure that there is no leakage of
349 information between training and validation/testing.

350 We trained a CNN with two input branches. The first branch took the haplotype
351 matrices as input and included three 1D-convolutional layers (kernel size = 2), each followed by
352 average-pooling. These layers contained 1250, 256, and 256 neurons, respectively. Each of these
353 layers uses dropout normalization (25%), L2-regularization of the weights (λ = 0.0001), and
354 ReLU activation functions. The second branch takes the position vector as input and contains
355 one densely connected layer with 64 neurons, again using dropout normalization (10%) and a
356 ReLU activation function. The two branches are then merged into another densely connected
357 layer of 256 neurons with ReLU activation functions. Finally, the output layer is a single neuron
358 with a simple linear activation function that predicts the continuous ρ value. The CNN was
359 trained using the Adam optimization algorithm, using mean-squared error as our loss function,
360 and a mini-batch size of 32. The CNN was trained for 16 iterations.

361 We compared our CNN's results to those of `LDhat` version 2.2a
362 (https://github.com/auton1/LDhat). We chose `LDhat` because it is widely used to estimate

363 historical recombination rates, and because it can be efficiently run on large data sets. LDhat will
364 estimate ρ only for a specified population mutation rate ($\theta = 4N\mu$), and we supplied it with the
365 exact θ value used for each coalescent simulation. This was done by creating five likelihood
366 lookup tables using the `complete` program, all set for 50 haploid chromosomes, for the
367 following θ values: 6, 12, 18, 24, and 60. Respectively, these correspond to N = 5,000, 10,000,
368 15,000, 20,000, and 50,000 (the same values we used for training our CNNs). `LDhat` only
369 predicts values within the bounds of the lookup table. Therefore, to facilitate a fair comparison to
370 results from our CNN, which is unbounded, we selected the maximum ρ value in the likelihood
371 lookup table to be 133.3% of the true maximum for each θ. We then set the grid size of ρ equal 1,
372 and estimated ρ on the test set using `LDhat`'s `pairwise` program.
373 In contrast, the CNN was not provided information about θ, and instead had to infer ρ
374 independent of θ. This ability would be a desirable property for an estimator, as θ is likely to vary
375 considerably across the genome and outside of simulated data sets one may never know θ
376 precisely. On the other hand, the CNN was provided with the physical distance between
377 segregating sites, information `LDhat` does not utilize but which will generally be available when
378 making inferences on real data. Both of these factors make our direct comparison of the CNN
379 with `LDhat` imperfect because each had access to information the other lacked when producing
380 its estimate. Nonetheless we consider this example a useful illustration of the CNN's power.
381
382 **Recombination rate: autotetraploid version**
383 We sought to train a CNN to estimate a locus-wide recombination rate in autotetraploid
384 genomes. To add a level of methodological realism to this problem, we did so from a matrix
385 storing a simple summary of read pileup information at each site for each individual.
386 To this end, we generated new coalescent simulations with 48 chromosomes each
387 following the procedure outlined above for the haploid CNN. This approach is reasonable
388 because it has been shown that the standard coalescent approximates the appropriate coalescent
389 for autotetraploids as long as N is larger than a few hundred (Arnold *et al.* 2012). We generated
390 217,500 coalescent simulations, and randomly assigned 200,000 to the training set, 10,000 to the
391 validation set, and 7,500 to the test set. Next, within each coalescent simulation, we randomly
392 partitioned our 48 chromosomes into twelve sets of four. Each set represents one synthetic
393 autotetraploid genome and every site has five possible genotypes (*AAAA*, *AAAa*, *AAaa*, *Aaaa*, and

394 *aaaa*). For each autotetraploid genome *i* and each site *j* we simulated the number of reads
395 covering the site (C_{ij}) by drawing a random sample from a Poisson distribution with $\lambda = 25$. Then
396 we selected the number of reads representing the *a* allele $R_{ij} \sim Binom(n=C_{ij}, p=x_{ij})$, where x_{ij}
397 represents the frequency of the *a* allele in the tetraploid genotype (i.e. 0, 0.25, 0.5, 0.75, and 1 for
398 the five genotypes listed above). For each individual *i* at site *j*, the corresponding entry in the
399 input matrix was the fraction R_{ij}/C_{ij}, i.e. the fraction of reads supporting the derived allele. The
400 *AAAA* and *aaaa* genotypes were always 0 and 1, respectively. For the three heterozygous
401 genotypes (*AAAa*, *AAaa*, and *Aaaa*), R_{ij}/C_{ij} varied based on sampling error but had expected values
402 of 0.25, 0.5, and 0.75, respectively. Thus at each site the original 48 chromosomes were reduced
403 to a set of 12 values corresponding to the fractions of reads supporting the *a* allele in a pool of
404 sequence reads from an autotetraploid sequenced at ~25X coverage. Note that this scheme does
405 not include sequencing error but is nonetheless adequate for our proof of concept.
406 As above, we sorted the rows of this matrix by genetic similarity and padded each matrix
407 with zeros to a length of 460 (the most segregating sites of any of the simulated matrices) before
408 transposing, yielding a 460×12 matrix. Again, we recorded the padded vector of positions from
409 the simulation output. Our CNN architecture was identical to the one given above for the phased
410 haplotype version, except for the dimensionality of the input changed to 460×12, and we
411 reduced the first convolutional layer from 1250 to 256 because of the smaller second dimension
412 of the input. The CNN was trained for 9 iterations.
413
414 **Detecting selective sweeps and discriminating between modes of selection**
415 For detecting selective sweeps, we used the same coalescent simulations that Schrider and Kern
416 (2017) used to train a classifier to detect sweeps in the JPT population (Japanese individuals from
417 Tokyo) from Phase 3 of the 1000 Genomes dataset (Auton *et al.* 2015). The JPT demographic
418 scenario is one where detecting selective sweeps is fairly difficult (see Figure S1 from Schrider and
419 Kern 2017), as expected for bottlenecked populations (Jensen *et al.* 2005). For this CNN, we
420 began with a set of 269,000 simulated genomic windows with the 5 following classes: a recent
421 hard sweep (i.e. fixation of a *de novo* beneficial mutation), a recent soft sweep (i.e. fixation of a
422 beneficial but previously neutral segregating polymorphism), a region linked to a nearby hard
423 sweep, a region linked to a nearby soft sweep, and a neutrally evolving region. Each simulated
424 alignment contained 208 chromosomes and we kept only coalescent simulations that contained ≤

425 5,000 segregating sites, and again padded with zeros so that all matrices were 208×5000. This
426 left 238,655 simulations, and from those we constructed a training set of 233,655 simulations. In
427 trial runs, we found that regions flanking hard and soft sweeps were the most difficult classes to
428 predict, so we shifted the balance of our training set so that is was comprised of approximately
429 13% neutral regions, 17% each for hard and soft sweeps, and 26.5% each for regions linked to
430 nearby hard and soft sweeps windows. We then set aside an evenly balanced set of 2,000
431 simulations for validation and 3,000 for testing.

432 As before, we sorted each matrix by genetic similarity among chromosomes and then
433 transposed the matrix to 5000×208. We also extracted the segregating site positions vector from
434 these simulations which were generated by `discoal` (Kern and Schrider 2016), which like `ms`
435 represents each position as a real number between zero and one.

436 As above, we trained a CNN with two input branches. The first branch took the
437 haplotype matrices as input and included five 1D-convolutional layers (kernel size = 2), each
438 followed by average-pooling. These layers each contained 256 neurons and used dropout
439 normalization (20%). The second branch takes the position vector as input and contains one
440 densely connected layer with 64 neurons, again using dropout normalization (10%). The two
441 branches are then merged into another densely connected layer of 256 neurons with 25%
442 dropout. Each hidden layer of the network used L2-regularization of the weights ($\lambda = 0.0001$)
443 and ReLU as the activation function. Finally, the output of this layer was fed to a five neuron
444 layer with softmax activation functions that predicts the five classes given above. The CNN was
445 trained using the Adam optimization algorithm, the categorical cross-entropy loss function, and a
446 mini-batch size of 32. The CNN was trained for 3 iterations.

447

448 **Inferring population size histories**

449 To show how CNNs can be used to infer species' demographic histories, and how CNN
450 architecture can impact this inference, we experimented with a variety of CNN approaches to
451 infer the 5 parameters of a 3-epoch model of instantaneous population size changes (i.e. 3
452 population sizes and 2 times of size change). We also use this challenging problem as an
453 opportunity to evaluate how alternative approaches to building a CNN can influence its
454 performance. In effect, we conducted a full grid search of the following attributes of both our
455 CNN architecture and input/output format: the dimensionality of our convolutions (1D or 2D),

456 the kernel size (i.e. the width of our 1D convolutional filters and both the height and width of our
457 square 2D filters; we tried each multiple of 2 raging from 2 to 10), whether to include dropout
458 (yes or no) following max pooling steps or dense layers, whether to sort our rows based on
459 similarity (yes or no), whether to log-scale our response variables (yes or no), and whether to
460 represent ancestral and derived alleles as -1/1 or as 0/255. When included, our dropout layers
461 immediately followed both max pooling steps, the dense layer following the distance input layer,
462 and the final dense layer. Each of these dropout steps randomly removed 25% of neurons. Each
463 response variable was transformed to a Z-score according to the sample mean and variance for
464 that variable across all simulated examples.

465 The network we used for this task had two branches: a standard CNN like that depicted
466 in Figure 1B–C but with more convolutional layers (four CNN layers each producing 128 filters
467 and each followed by a max pooling layer with a kernel size of 2), and a dense neural network
468 layer (consisting of 32 nodes) taking positional information as its input, and concatenating its
469 output with that of the final max pooling layer of the CNN prior to being fed into the final dense
470 layer (256 nodes). The positional information was a vector, $\boldsymbol{d}$, whose length was the maximum of
471 the number of segregating sites observed across all simulated examples minus one. Each value in
472 the vector d_i was simply the distance (scaled between zero and one where one is the total length of
473 the simulated region) between segregating site i and site i-1.

474 In total, we simulated 100,000 examples using `ms`. 10,000 each were set aside for testing
475 and validation, while the remaining 80,000 were used for training. The simulated population size
476 histories were generated randomly—each demographic model parameter was drawn uniformly
477 from a rage listed in Table S1. Each simulated region was roughly equivalent 1.5 Mbp in the
478 human genome, assuming per base pair mutation and recombination rates of 1.2×10^{-8} and
479 1×10^{-8}, respectively. However, in order to make the size of the simulation output more tractable
480 for processing in a CNN we divided the mutation rate by 10 (equivalent to randomly
481 downsampling the number of polymorphisms included in the input by a factor of 10). During
482 training we used a batch size of 200, trained our networks for up to 10 iterations, and retained
483 the best performing CNN as assessed on the validation set. Often the best CNN was obtained
484 prior to completing all 10 training iterations. We then evaluated the performance of the best
485 CNN for each network architecture and input format on the test set by calculating total RMSE

(our loss function for this task); we also calculated Spearman correlation coefficients between the true and predicted values for each of the five demographic model parameters.

RESULTS

Our goal is to use a CNN to make population genetic inferences from an alignment image, which can be thought of as matrices where each entry represents the allele present in a given chromosome at a given site. In particular, we focus on four distinct problems: identifying local introgression, estimating the recombination rate, detecting selective sweeps, and inferring population size changes. We chose these four tasks because each represents a different challenge in population genetic inference, each with its own attendant branch of theory. To show the ability of CNNs to solve problems that are not already solved, we extended our recombination inference to infer recombination rates in autotetraploids with tetrasomic inheritance. We took

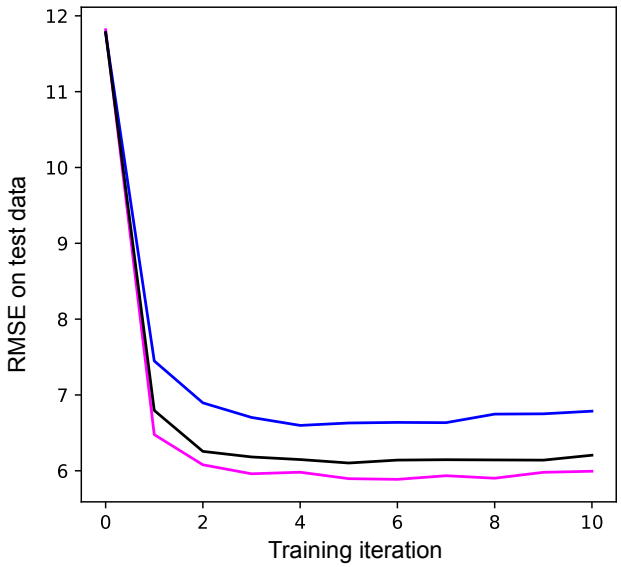

Figure 3: The impact of input data reorganization on accuracy. We show the root mean squared error (RMSE) of a 1D CNN's predictions of θ as assessed on 1,000 test alignments after a given number of training iterations. Each line is the average of 10 runs. The blue line shows accuracy after training using alignment matrices with each row representing one chromosome. The black line shows accuracy after transposing of all matrices so that chromosomes correspond to columns (so that 1D convolutional filters examine each individual at a group of adjacent segregating sites). The magenta line shows the impact of transposing matrices, and sorting the chromosomes in the alignment matrix by genetic similarity.

498 slightly different approaches in training the CNN for each task, though we hope any lack of
499 consistency in part demonstrates the different ways in which one can successfully implement and
500 tune CNNs.

501 Below, we address each of these problems in turn, providing a brief overview of the
502 phenomenon in question and existing methodology before describing our results using CNNs.
503 But prior to tackling these problems, we begin by introducing a few strategies for reorganizing
504 our input data that we found to be helpful in making CNNs work efficiently with population
505 genetic alignments.
506

507 **Using a CNN to make inferences from an alignment: a simple test case**

508 We evaluated the performance impact of transposing the alignment matrix and sorting the
509 chromosomes in the alignment matrix by genetic similarity. We did this using a CNN trained to
510 estimate the population-scaled mutation rate, θ, in an equilibrium population. We found that
511 both of these techniques accelerates the decline in root-mean-square error RMSE (Figure 3).
512 Transposing the alignment matrix so that chromosomes are represented by rows and
513 polymorphisms by columns has a particularly notable effect (compare blue and black lines in
514 Figure 3). Additionally, sorting the chromosomes by genetic similarity further increases the
515 accuracy of the CNN when combined with the matrix transposition above (magenta line);
516 alternatively, using a permutation-invariant network architecture would abdicate any need for
517 this step (Chan *et al.* 2018). The effect of transposition should disappear when using 2D
518 convolutions because in those cases we always used a square convolutional filter matrix
519 (Methods), but we found that 1D CNNs often performed as well as 2D CNNs (data not shown).
520 Thus, unless otherwise specified we use 1D convolutions for the tasks discussed below.
521

522 **CNN's can accurately detect introgressed loci**

523 Recent studies indicate that closely related species often exchange genes (Kulathinal *et al.* 2009;
524 Martin *et al.* 2013; Brandvain *et al.* 2014; Fontaine *et al.* 2015). There are several motivations for
525 locating genomic segments introgressed from one species into another. For one, the occurrence
526 of cross-species gene flow raises the possibility of adaptive introgression, wherein a beneficial
527 allele enters a population via migration from a related species (reviewed in Hedrick 2013).
528 Discovering introgressed loci can therefore identify alleles underlying rapid ecological adaptation

529 as well as the source of these alleles. In addition, uncovering genomic regions that are and are not
530 porous to cross species gene flow may help to illuminate the genomic basis of reproductive
531 isolation (Turner *et al.* 2005).

532 Researchers have thus sought to devise methods capable of detecting introgressed regions
533 from multispecies population genomic data sets. These include methods that attempt to infer the
534 ancestry for each individual at each site (e.g. Price *et al.* 2009; Lawson *et al.* 2012; Sohn *et al.* 2012)
535 and those that explicitly seek to discriminate between introgressed and non-introgressed loci
536 (Geneva *et al.* 2015; Rosenzweig *et al.* 2016; Schrider *et al.* 2018). We trained a CNN to identify
537 introgression in a scenario modeled after *Drosophila simulans* and *D. sechellia* (Methods), two species
538 where there is evidence for recent gene flow (Garrigan *et al.* 2012). Figure 4 displays the results of
539 these tests in the form of confusion matrices, which show the fraction of test examples correctly
540 predicted for each class (diagonal values) as well as the fractions incorrectly assigned (off-diagonal
541 values). Overall, this CNN classified 88.5% of test simulations correctly. The most difficult

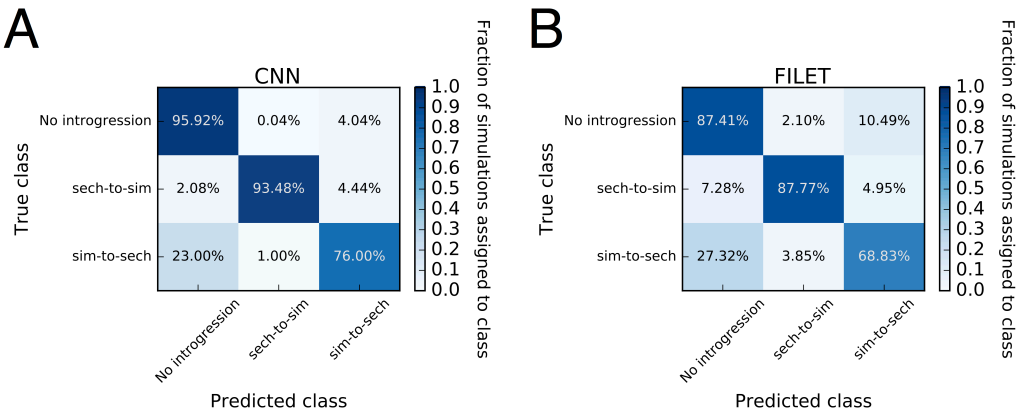

Figure 4: Performance of classifiers for detecting introgression. We use confusion matrices to show the performance of a CNN trained to detect genomic regions of introgression between two closely related species (panel A), and a competing method that uses a vector of summary statistics to the same end (FILET; panel B). These classifiers were tested on data simulated under a joint demographic history inferred from a sample of *Drosophila simulans* and *D. sechellia* individuals (as described in the Methods) with and without introgression. The classifiers seek to discriminate among three classes: no introgression in the genomic window being examined, introgression from *D. sechellia* to *D. simulans*, and introgression from *D. simulans* into *D. sechellia*. Each entry in the matrix shows the fraction of test examples belonging to the class specified on the *y*-axis that were inferred by the method to belong to the class specified on the *x*-axis. Correct classifications are those found along the diagonals, while all off-diagonal entries represent incorrect classifications.

542 scenario for the CNN was introgression from *D. simulans* into *D. sechellia*, which it misclassified as
543 "no introgression" 23% of the time. For the other two classes the CNN accuracy was >95%.
544 Importantly, in every case this CNN achieved greater accuracy than FILET, a machine learning
545 approach that leverages a vector of 31 summary statistics (Schrider *et al.* 2018). Importantly,
546 FILET was previously shown to outperform several competing methods, including two statistics
547 for detecting introgression (Joly *et al.* 2009; Geneva *et al.* 2015), and a tool that infers local
548 ancestry tracks for each individual (Lawson *et al.* 2012).
549
550 **Estimating historical recombination rate**
551 Recombination creates new combinations of alleles, and the degree of linkage between selected
552 sites affects the efficiency with which natural selection can act on each individual site (Hill and
553 Robertson 1966). The interplay of selection and recombination also influences the landscape of
554 diversity across the genome (Begun and Aquadro 1992). Knowledge of recombination rates is
555 thus key to population genetics research. As a more practical alternative to estimating rates
556 directly (e.g. from pedigrees; Kong *et al.* 2010), one can infer recombination rates from
557 population genetic data by examining associations among alleles at different sites. A number of
558 methods have been proposed to solve this problem, including summary statistic estimation
559 approaches (e.g. Hudson and Kaplan 1985; Hudson 1987; Hey and Wakeley 1997), composite
560 likelihood-based methods (e.g. Hudson 2001; McVean *et al.* 2004; Chan *et al.* 2012), and machine
561 learning tools using a vector of statistics (Lin *et al.* 2013; Gao *et al.* 2016). We sought to determine
562 if a CNN taking an alignment image as input could be trained to tackle this task.
563 To address this problem, we first trained a CNN to estimate the historical population
564 recombination rate $\rho=4Nr$ (where r is the crossover rate per base pair per meiosis) from phased
565 chromosomes. This is the simplest scenario, as the arrangement of alleles on chromosomes is
566 completely resolved. Following training, we compared the CNN's performance to that of `LDhat`
567 (McVean *et al.* 2004), a widely used composite likelihood method, on the same testing data
568 (Figure 5). We generated a test set of alignments whose values of ρ spanned three orders of
569 magnitude, from 0.0002 to 0.2 (expressed per bp). Overall, both approaches performed well at
570 predicting the true value of ρ. LDhat had an R^2 = 0.77 and an RMSE = 0.016, whereas the CNN
571 had a R^2 = 0.86 and an RMSE = 0.011 (Figure 5A,B). `LDhat` appears to estimate ρ slightly
572 better than the CNN for lower recombination rates, whereas the CNN performs better at the

573 higher values of ρ (Figure 5C). Additionally, the CNN appears to provide a roughly unbiased
574 estimator of ρ, while `LDhat`'s estimates appear downwardly biased.
575 Because the CNN was capable of estimating ρ independent of θ, we were interested to see

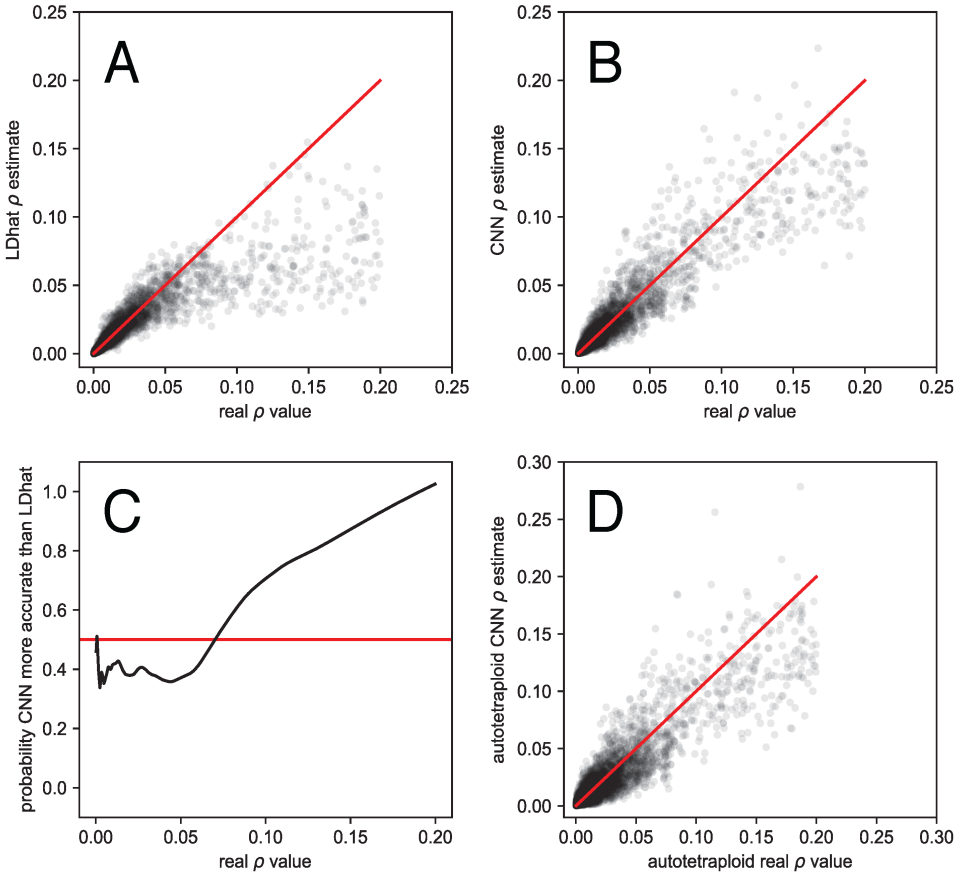

Figure 5: Accuracy of recombination rate estimates from LDhat and our CNN. Panels A and B show the real ρ values on the *x*-axes and LDhat's (A) and the CNN's (B) predictions on the *y*-axes. Panel C again shows the real ρ values on the *x*-axis, and the probability that the CNN was more accurate than LDhat (black line) on the *y*-axis. This probability was calculated by scoring estimates where the CNN outperformed LDhat as one and the reciprocal as zero, and then smoothing these values with a lowess curve with a span of 15%. The red line represents the expectation if both methods had identical accuracy. Panel D shows the results from the simulated autotraploid model, with the real ρ values on the *x*-axes and the CNN prediction on the *y*-axes.

576 how well it could interpolate between the θ values it was trained with. The CNN was trained with
577 a large gap between $\mathcal{N}$ = 20,000 and $\mathcal{N}$ = 50,000 (and thus a large gap in θ; see Methods), so we
578 used coalescent simulations to generate an additional test set with $\mathcal{N}$ values drawn uniformly

among 30,000, 35,000, 40,000, and 45,000. When tested on these data the CNN's predictions had an $R^2 = 0.82$ and an RMSE = 0.017. This represents a slight decrease in accuracy from the values obtained when tested on the same N values used in training, but nonetheless shows that the CNN can interpolate between training parameters without a dramatic loss in accuracy. This could be a useful property, for example in cases where N (or θ) is unknown, but where one can generate coalescent simulations across a range of plausible values.

Further complications arise when estimating ρ from unphased data. Under this scenario the arrangement of alleles on chromosomes is not known. One work-around is to first phase the alleles and then infer ρ as above, but not all data sources are easily phased, and phasing errors will, of course, reduce accuracy. Another approach is to analyze the unphased data directly. The relevant theory required to tackle this problem in a probabilistic manner has been worked out for unphased diploids (Auton and McVean 2007), but expanding this theory to higher ploidies would require a substantial effort. Take for example an autotetraploid with tetrasomic inheritance, where there are five possible genotypes (*AAAA*, *AAAa*, *AAaa*, *Aaaa*, and *aaaa*). To further complicate things, after sequencing an autotetraploid genome to a moderate depth of coverage and identifying polymorphisms, the true underlying genotype may be uncertain. For example, given a site with 10 reads supporting *A* and 10 supporting *a*, the true genotype could be *AAAa*, *AAaa*, or *Aaaa*. Perhaps the greatest strength of likelihood-free approaches like CNNs is that they can be used to solve problems for which no appropriate tool exists, and can be easily customized to accommodate methodological and biological realities by constructing appropriate training sets. To show the utility of CNNs in addressing novel population genomic inference problems, we designed a CNN capable of inferring ρ from a simulated set of sequence reads from an unphased autotetraploid population sample.

We used a simple simulation scheme to produce read counts for each allele at each site for each individual in a sample of 12 autotetraploids, each with approximately 25X expected genome-wide coverage (see Methods). Rather than allelic assignments, the input matrix for this CNN contains for every site in each individual the fraction of reads bearing the *a* allele. Deriving a likelihood function for ρ under this formulation may be possible, but such a solution has not yet been attempted. However, appropriately designed artificial neural networks are universal approximators, meaning that they have the potential to approximate any continuous function over a compact input space (Hornik 1991). Thus it is possible for a CNN to approximate the

610 desired likelihood function, even in its absence. To this end we trained a CNN with a similar
611 architecture to the one used above on phased haploid chromosomes (see Methods). We evaluated
612 the performance of this CNN a set of simulations where ρ again ranged from 0.0002 to 0.2 (still
613 scaling by $4N$, rather than $8N$ which would be appropriate for tetraploids, so the result can be
614 compared to those above). The CNN's predictions had an R^2 = 0.83 and an RMSE = 0.012
615 (Figure 5D). As before, the estimate of ρ was made independent of θ, which varied over an order
616 of magnitude. The fact that this autotetraploid network performed only slightly worse than the
617 haploid version demonstrates that in addition to outperforming current methods, CNN's can also
618 solve a problem for which no model-based likelihood (or even composite likelihood) approach
619 has been obtained.
620

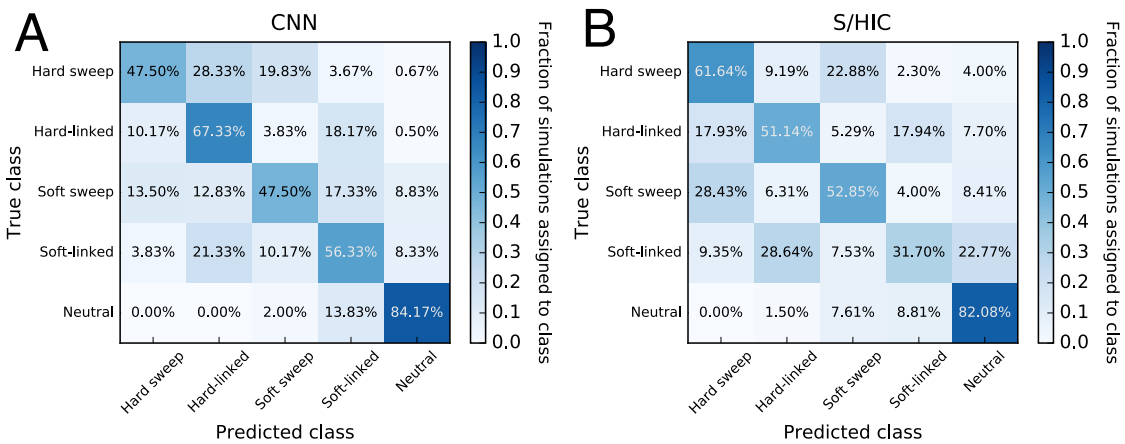

Figure 6: Confusion matrices showing accuracies of two methods that seek to detect recent positive selection by discriminating among hard sweeps, soft sweeps, unselected regions closely linked to hard and soft sweeps, and neutrally evolving regions. (A) Confusion matrix summarizing the performance of our CNN, which uses an alignment image as input. (B) Performance of S/HIC, which uses a vector of summary statistics each measured in windows surrounding the region to be classified.

621 **CNNs can accurately detect and categorize signatures of recent positive selection**
622 When a new mutation is immediately favored by positive selection, it rapidly increases in
623 frequency until it fixes (i.e. completely replaces all other alleles at that site). This phenomenon,
624 referred to as a hard selective sweep, drastically reduces the amount of linked neutral variation
625 (Maynard Smith and Haigh 1974), and produces characteristic skews in the allele frequency
626 spectrum (Fay and Wu 2000) and linkage disequilibrium at linked sites (Kim and Nielsen 2004).

627 Alternatively, in a process known as a "soft sweep" populations may adapt via selection on a
628 polymorphism that has been segregating for some time, such that the adaptive allele exists on
629 numerous haplotypes (Hermisson and Pennings 2005). In order to uncover the genomic regions
630 underlying recent adaptation, a large number of methods have been devised to detect and
631 characterize selective sweeps. These include summary statistics (Kelly 1997; Fay and Wu 2000;
632 Kim and Nielsen 2004; Voight *et al.* 2006; Garud *et al.* 2015), composite likelihood-based
633 approaches (Kim and Stephan 2002; Kim and Nielsen 2004; Nielsen *et al.* 2005; Vy and Kim
634 2015), and supervised machine learning approaches using a vector of statistics to obtain greater
635 power than individual tests/statistics (Lin *et al.* 2011; Pybus *et al.* 2015; Schrider and Kern 2016;
636 Sheehan and Song 2016; Sugden *et al.* 2018). Although these efforts have led to considerable
637 progress, detecting and distinguishing between hard and soft sweeps remains a major challenge.
638 We built a CNN to detect selective sweeps and to discriminate between hard sweeps and
639 soft sweeps. This CNN follows the S/HIC method of Schrider and Kern (2016) by casting the
640 problem as a classification task where the genomic region being examined is assigned to one of
641 five disjoint classes: a recent classic "hard" sweep, a recent "soft" sweep, a region linked to a
642 nearby hard sweep, a region linked to a nearby soft sweep, or a neutrally evolving region.
643 However, rather than adopting S/HIC's approach of using a large vector of statistics, the CNN
644 takes an alignment image as input. We tested both methods against data simulated under a
645 challenging demographic history estimated from human population data (Methods). As
646 evidenced by the confusion matrices in Figure 6, the CNN has higher overall accuracy than
647 S/HIC (60.5% versus 55.9%), a remarkable feat given that S/HIC was shown to have greater
648 power than a number of competing methods (Schrider and Kern 2016). While S/HIC appears to
649 be somewhat more sensitive to sweeps, the CNN is achieves a nearly 4-fold reduction in false
650 positive rate: 2% of neutral simulations are classified as sweeps by the CNN, versus 7.61% for
651 S/HIC. This quality may be particularly desirable when scanning genomes where sweeps are
652 relatively rare and thus a high degree of specificity is required to maintain a low false discovery
653 rate. Note that these classifiers were both trained under the same demographic history from
654 which the test data were generated. We would not expect this CNN to match S/HIC's
655 robustness to demographic misspecification given that S/HIC's feature vector was designed with
656 this in mind, though we did not test this. Nonetheless, the fact that the CNN has greater accuracy
657 than S/HIC under this difficult test scenario is highly encouraging.

658

659 **CNNs can extract demographic information from alignments**

660 A major focus of population genetics research is to use genomic data to elucidate species'
661 demographic histories—the extent and timing of population size changes, and the history of
662 population splits and migration events. For example, a host of population genetic approaches
663 have been devised to infer the times and intensities of population contractions and expansions
664 over the course of a species' recent history (e.g. Marth *et al.* 2004; Schiffels and Durbin 2014; Liu
665 and Fu 2015), and to elucidate the history of population splits and subsequent gene flow (Nielsen
666 and Wakeley 2001; Hey 2009), and population merging events (e.g. Lipson *et al.* 2013; Loh *et al.*
667 2013). We sought to ask whether CNNs can effectively extract demographic information from
668 alignment images, focusing on the task of inferring population size histories. In particular, we
669 attempted to train a CNN to estimate the parameters of a three-epoch model of instantaneous
670 population size changes. There are five such parameters: the ancestral population size (N_2), the
671 time of the more ancient population size change (T_1), the population size after this change (N_1),
672 the time of the more recent change (T_0), and the present-day population size (N_0); our response
673 variable is the vector of these 5 real-valued parameters. Thus this analysis also allows us to assess
674 the ability of CNNs to predict multiple population parameters simultaneously.

675 We simulated 50 haploid chromosomes under a variety of randomly selected population
676 size histories, and trained a CNN to estimate the demographic model parameters. The simulated
677 region was roughly equivalent in length to 1.5 Mbp of the human genome (Methods). Because
678 we found this problem to be comparatively difficult, we experimented with a variety of
679 hyperparameters governing the neural network structure and input/output format. In Table S2
680 we show the optimal RMSE (i.e. the minimum RMSE across training iterations) for each
681 hyperparameter combination examined. This experiment revealed several general trends. First,
682 1D convolutional networks tended to fare slightly better than their 2D counterparts (median
683 RMSE of 0.52 across all hyperparameter combinations with 1D convolutional filters, and
684 median RMSE 0.54 for 2D convolutions; $p=1.1\times10^{-4}$; Mann-Whitney *U* test); however several
685 2D networks performed nearly as well as the best 1D network, achieving an RMSE of <0.5 while
686 the best score obtained overall was 0.43. Second, smaller convolutional filters tended to perform
687 slightly better than larger ones—we observed a positive correlation of kernel size with RMSE
688 across hyperparameter combinations ($\rho=0.26$; $p=6.9\times10^{-4}$; Mann-Whitney *U* test). For example,

the median validation RMSE was 0.51 for a kernel size of 2 versus 0.56 for a kernel size of 10. Third, log-scaling the demographic parameters to be estimated increased accuracy (RMSE decreased from 0.55 to 0.52; p=0.020; Mann-Whitney U test). For this problem sorting chromosomes by relatedness resulted in a small improvement (RMSE decreased from 0.54 to 0.53; p=0.034). Encoding ancestral and derived alleles as '0' and '255' (i.e. black and white in a grayscale image), respectively, versus '-1' and '1' had a significant influence on accuracy, with the former yielding better performance than the latter (RMSE of 0.51 vs. 0.60; $p=1.5\times10^{-15}$). Finally, using dropout resulted in a slight decrease in accuracy (median RMSE increased from 0.52 to 0.55) though this was not statistically significant (p=0.092). We note that these trends may change if the amount of training data is increased or decreased, and may not necessarily hold for other tasks.

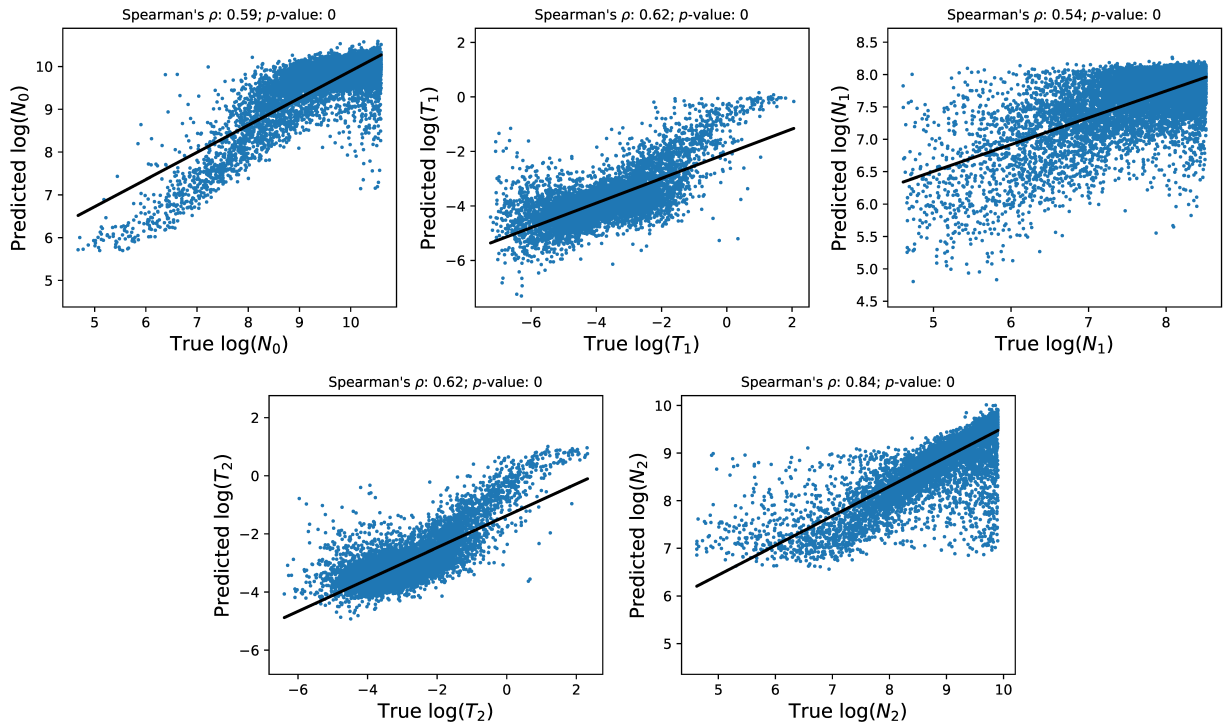

Figure 7: Accuracy of demographic inference CNN. The scatterplots show the correlation between true and predicted demographic parameter values using our best-performing CNN for this task when applied to an independent test set.

In Figure 7, we show the correlation between the true and inferred values for each of these 5 parameters for the best performing network. For N_0 and T_0, these correlations are quite high, implying that our CNN can recover the true values reasonably well. However, for the

704 remaining parameters, the correlation is lower (though still highly significant), and our CNN
705 produces downwardly biased estimates when the values of these parameters are larger. Although
706 our accuracy is far from perfect, we consider these results fairly encouraging because we are only
707 examining a single moderately sized genomic region, while other modern demographic inference
708 methods use data from across the genome. For example, ∂a∂i (Gutenkunst *et al.* 2009) uses allele
709 frequencies measured at a large number of polymorphisms (e.g. those found in all distal
710 intergenic regions across the genome; Gazave *et al.* 2014). PSMC andMSMC (Li and Durbin
711 2011; Schiffels and Durbin 2014) take data from a single very large recombining region such as
712 an entire chromosome. In essence, we are currently only able to utilize information about the
713 coalescent histories of the region in question—and this collection of histories may not match that
714 of the entire population, which would be more accurately reflected in genome-wide data. In the
715 Discussion, we address prospects for incorporating genome-scale data in demographic inference.
716

717 **DISCUSSION**

718 **Convolutional neural networks are well suited for population genetic problems**

719 Population geneticists have devised a wide array of computational methods to make evolutionary
720 inferences from genomic data. Typically the goal of these efforts is to aggregate information
721 across genomic sites in order to make an accurate inference. These methods include maximum
722 likelihood approaches (e.g. Kim and Stephan 2002; Nielsen *et al.* 2005; Gutenkunst *et al.* 2009;
723 Liu and Fu 2015), probabilistic graphical models such as hidden Markov models (e.g. Turner *et*
724 *al.* 2005; Boitard *et al.* 2009; Lawson *et al.* 2012), and those rely on the use one or more summary
725 statistics designed to characterize patterns of variation within a genomic region (e.g. Tajima
726 1989; Fu and Li 1993; Kelly 1997; Fay and Wu 2000; Kim and Nielsen 2004; Voight *et al.* 2006;
727 Ferrer-Admetlla *et al.* 2014). While these approaches differ substantially from one another, they
728 all have one thing in common: they make use of population genomic theory to connect the
729 features of a data set to the underlying evolutionary process. Here we have demonstrated the
730 potential of an alternative approach: treating population genetic inference as an image
731 recognition problem where the "image" is the population genetic alignment, which is directly fed
732 as input to a convolutional neutral network. In contrast to mainstream approaches, this CNN
733 approach makes use of the entirety of the data and does not rely on results from population
734 genetic theory.

735 Here we have shown that CNNs perform remarkably well on a number of problems in
736 population genetics. We developed CNNs with greater power to detect selective sweeps, identify
737 introgressed loci, and infer local recombination rates when compared to current methods on
738 simulated data sets. The CNNs for detecting sweeps and introgression demonstrate the ability to
739 use an alignment image to distinguish among multiple evolutionary models, while the
740 recombination rate estimator demonstrates that continuous parameters can also be inferred.
741 Finally, although our demographic parameter estimates were fairly imprecise, they were only
742 based on a short stretch of the genome, and nonetheless demonstrate that CNNs have the
743 potential to infer multiple parameters from a sequence alignment. While we were in the process
744 of preparing this manuscript, Chan et al. completed an important study demonstrating that a
745 CNN can accurately detect recombination hotspots (Chan *et al.* 2018). Taken together these
746 results suggest that CNNs have enormous potential as a general paradigm for population genetic
747 inference.

748 The effectiveness and generality of CNNs in population genetic inference should not be
749 surprising. CNNs offer a number of intrinsic advantages that make them particularly amenable
750 to population genetic data. First, there have been a number of efforts to move in the direction of
751 making inferences on the basis of the full complement of data present in an alignment rather
752 than one or more summary statistics (Li and Stephens 2003; Lawson *et al.* 2012; Smith *et al.*
753 2018). CNNs represent a natural way of examining the entirety of an alignment in order to
754 increase inferential power. The development of novel CNN architectures to better handle spatial
755 associations in the data across multiple scales (Yu and Koltun 2015) has the potential to improve
756 CNN-driven population genetic inference even further; for example, improved ability to detect
757 both the localized reduction in diversity at a sweep (Maynard Smith and Haigh 1974) as well as
758 the potentially confounding skews in patterns of diversity produced in its flanking regions
759 (Schrider *et al.* 2015) would be beneficial in sweep detection.

760 Another desirable property of CNNs is that they effectively perform automated feature
761 detection (LeCun *et al.* 2015). Because they discover discriminatory information directly from the
762 image, there is no need to manually construct an optimal set of features. This brings up perhaps
763 the strongest quality of CNNs in the context of evolutionary inference: because CNNs are not
764 rooted in population genetic theory, they can make predictions for phenomena for which there
765 exists no analytical expectation.

766 Indeed, CNNs can tackle problems for which no relevant summary statistics have been
767 devised—vectors of such statistics are required for other likelihood-free methods such as ABC
768 (Beaumont 2010) or more traditional supervised machine learning techniques (Schrider and
769 Kern 2018). On a related note, neural networks are particularly amenable to the incorporation of
770 disparate data types with no prior knowledge of their relationships. For example, here we have
771 included both genotype information and positional information for segregating sites as branches
772 to our networks, allowing for both to be used together in prediction despite the fact that our
773 network does not "know" how these two pieces of information relate to one another. All that is
774 required is appropriate training data. Thus, we may not have to wait for theoretical advances in
775 order to draw inferences from data, provided we are concerned with evolutionary models for
776 which training data can be obtained from simulation—including the wide range of scenarios that
777 could potentially be investigated via increasingly flexible and efficient forward simulators
778 (Thornton 2014; Haller and Messer 2017; Kelleher *et al.* 2018)—or otherwise.
779 This point is driven home by the remarkable success of our CNN that estimating
780 recombination rates in autotetraploids from read pileup information alone—despite the input's
781 lack of genotype calls, let alone phased haplotypes, these inferences are nearly as accurate as
782 those that we obtained from haplotype alignments. This result also suggests that CNNs may be
783 well suited for other inferences where genotype calls are unreliable (e.g. low coverage sequencing
784 data; Korneliussen *et al.* 2014) or unobtainable (e.g. pooled population sequencing; Schlötterer *et*
785 *al.* 2014). Given CNNs' remarkable flexibility, future studies should evaluate their potential to
786 tackle not only those problems examined in this paper, but the myriad additional important
787 challenges in evolutionary genetics to which they could be readily applied, including but not
788 limited to uncovering adaptive introgression (Racimo *et al.* 2016), joint inference of selective and
789 demographic histories (Sheehan and Song 2016), and even inferring structured outputs such as
790 ancestral recombination graphs (Rasmussen *et al.* 2014).
791
792 **To what extent are CNNs robust to model misspecification?**
793 Another particularly encouraging result of our recombination rate estimation analysis is that we
794 were able to infer rates for data generated from a range of parameter values to which the CNN
795 had not been exposed during training with very little decrease in accuracy. This ability to
796 interpolate between training values is a particularly desirable property. First, it implies that

797 CNNs can be used to create flexible inference tools using a modest training data set, and second
798 that researchers can focus training between reasonable parameter bounds, without knowing the
799 true (and often unknowable) underlying parameters; future efforts must explore the possibility of
800 training networks to be robust to more extreme cases of model misspecification.

801 One illustrative example of the potential pitfalls of model misspecification is the problem
802 of detecting selective sweeps without accounting for confounding demographic events. For
803 example, population bottlenecks will skew genealogies in a manner similar to sweeps (Simonsen *et*
804 *al.* 1995), and thus may result in a large fraction of false positives (Jensen *et al.* 2005; Nielsen *et al.*
805 2005). Schrider and Kern (2016) were able to mitigate this problem by designing a feature vector
806 that is sensitive to the spatial skews in patterns of variation created by a sweep but insensitive to
807 genome-wide skews produced by demographic events. Although this strategy is not possible with
808 CNNs because they perform automated feature extraction, it may be that incorporating training
809 examples generated under potentially confounding scenarios could alleviate this issue.

810 Therefore, future work must thoroughly 1) assess how CNNs trained under one range of
811 evolutionary parameters fare when applied to different parameterizations, and 2) determine
812 whether robustness to such misspecification might be achieved by training a CNN under a wide
813 range of parameter values that are likely to encapsulate the correct values—the recombination
814 rate estimator's successful interpolation suggests that this may be a possibility.

815

816 **Outstanding practical challenges associated with the application of CNNs to**
817 **sequence data**

818 Although the CNN approach outlined above has great potential, there are several outstanding
819 challenges with applying CNNs to a wider spectrum of problems. One important obstacle is the
820 large amount of training data required by CNNs, which makes applications requiring alignments
821 of large regions (e.g. entire chromosomes) more difficult. This challenge includes both the
822 generation of large labeled training examples, and time- and memory-efficient training with these
823 large examples given limited computational resources. Fortunately, continued improvements in
824 simulation speed (Kelleher *et al.* 2016; Kelleher *et al.* 2018) and the efficiency of CNN training
825 (Chilimbi *et al.* 2014; Yu and Koltun 2015; Jouppi *et al.* 2017; Köster *et al.* 2017) is mitigating this
826 problem. Such advances would be a boon for efforts to infer demographic parameters, which
827 require simultaneously examining data sampled from across the genome or along an entire

828 chromosome, unlike scans to infer locus-by-locus histories of
829 selection/recombination/introgression. Advances in handling large or high-resolution images
830 may also prove fruitful. For example, CNN-based strategies that simultaneously examine a
831 number of smaller "patches", each covering a portion of the image rather than the entirety of the
832 image (e.g. Lu *et al.* 2015), may aid efforts to extract demographic information from genome-
833 scale data.

834 Another challenge with the application of CNNs is that their performance can be
835 sensitive to network architecture (Szegedy *et al.* 2015). There is no underlying theory for selecting
836 optimal network architecture, though improved architectures are sure to continue to arise.
837 Though we uncover some promising CNN architectures for population genetic inference, we
838 suspect that substantial improvements can still be made.

839 We have also demonstrated that CNNs are sensitive to the input format of the population
840 genetic alignment, and our work has yielded several insights along this front. First, we found that
841 the ordering of haplotypes within the alignment can impact accuracy, and our results suggest that
842 it is often beneficial to reorder haplotypes so that more similar chromosomes appear next to one
843 another. This may be a suboptimal solution, and more creative approaches may be required to
844 provide a more general strategy. To this end, research into permutation-invariant neural
845 networks (Zaheer *et al.* 2017) may prove promising when dealing with sequence alignments. This
846 is evidenced by Chan et al.'s recent findings that a permutation-invariant architecture improves
847 both training speed and final accuracy of their CNN for detecting recombination rate hotspots
848 (Chan *et al.* 2018). We also observed that, somewhat surprisingly, 1D convolutions in the proper
849 orientation perform as well as the more widely used 2D convolutions in many cases. Scaling of
850 response variables for regression problems (both log-scaling and standardization) may also affect
851 accuracy. We therefore recommend that users experiment with these different ways of
852 representing their data, as well as different CNN architectures, in order to find the design that
853 works best for the task at hand.

854 Another important consideration of CNNs is that once trained, they are specialized to a
855 particular problem as defined by the training set. That is, a CNN trained to infer recombination
856 rates under a European demographic history may have reduced accuracy when applied to an
857 African sample. Training under a variety of demographic scenarios may make a CNN more
858 robust to this problem, but a question for further study is whether this can be accomplished

without a loss in power relative to a more specialized CNN. Even a change as subtle as adding another chromosome to a dataset will make one of our previously trained CNNs inapplicable, as the input matrix would no longer be the proper size and either a new CNN must be trained or the data subsampled. Importantly, Chan et al. (2018) describe an architecture that can allow for variation in the number individuals in the input matrix. In spite of these limitations, recent advances have greatly simplified training CNNs, and it will often be practical—or even preferable—for a researcher to create a CNN tailored to their specific data set.

Are CNNs a black box?

Unlike most existing population genetic inference methods, a CNN is not rooted in evolutionary models or summary statistics. Instead a CNN is an algorithm that tries to maximize its predictive accuracy by optimizing its internal logic operations on training data. Its lack of reliance on theory gives CNNs enormous flexibility; however, one consequence of this is that CNNs are in some ways a "black box". For example, a CNN cannot "explain" why it made a particular prediction given its input. Supervised machine learning algorithms in general have perhaps been unfairly maligned with this "black box" label. These methods can in principle reveal much about underlying processes by revealing which features are most informative under certain scenarios (i.e. feature ranking; see Breiman 2001). For example, the observation that certain features are highly informative for recent but not ancient introgression (Schrider *et al.* 2018) suggests some key differences between the genealogies produced under these two scenarios. It is even possible for supervised machine learning methods relying on feature vectors to reveal which features had the greatest influence on a particular prediction (e.g. Baehrens *et al.* 2010). A population geneticist could thus use supervised machine learning not only to find a soft selective sweep, but also to quickly reveal which summary statistics (and therefore genealogical properties) were most consistent with a soft sweep. Due to their complex inner workings, less progress has been made in breaking through the CNN "black box" as compared to more traditional supervised machine learning techniques. However, some successful explanatory tools are available for CNNs (Ribeiro *et al.* 2016), and there is ongoing research in this area. Moreover, because the CNN framework we adopt here works on images, it may be possible to translate future breakthroughs in CNN interpretation from other fields (e.g. image recognition) into population genetic inference. Thus a more optimistic view is that as CNNs and related methods become more interpretable, these

890 theory-free image recognition approaches may help to reveal theoretical insights into
891 evolutionary processes.

892 In the near-term, CNNs may remain useful only as a predictive tool, and we will continue
893 to rely on theoretical advances to improve our understanding of population genetic processes. In
894 spite of the shortcomings noted above, the highly encouraging results that we have laid out here
895 suggest that CNNs are able to discover information about the underlying genealogies from
896 alignment images and to use this information to more accurately elucidate the evolutionary
897 phenomena that have shaped these genealogies. CNNs thus have enormous potential for
898 population genomic inference. We believe that progress on a host of problems could accelerate
899 appreciably were this technology to be embraced by the field. Indeed, when it comes to the
900 business-end of population genetics—drawing accurate evolutionary inferences from data—we
901 predict that increasingly, theory-free approaches such as the ones we have describe here will
902 prove most effective at solving existing problems, and expand the universe of problems that
903 researchers can investigate.

904

905 **REFERENCES**

906

907 Achaz, G., 2009 Frequency spectrum neutrality tests: one for all and all for one. *Genetics* **183:**
908 249-258.
909 Arnold, B., K. Bomblies and J. Wakeley, 2012 Extending coalescent theory to autotetraploids.
910 *Genetics* **192:** 195-204.
911 Auton, A., L. D. Brooks, R. M. Durbin, E. P. Garrison, H. M. Kang *et al.*, 2015 A global
912 reference for human genetic variation. *Nature* **526:** 68-74.
913 Auton, A., and G. McVean, 2007 Recombination rate estimation in the presence of hotspots.
914 *Genome Res.* **17:** 1219-1227.
915 Baehrens, D., T. Schroeter, S. Harmeling, M. Kawanabe, K. Hansen *et al.*, 2010 How to explain
916 individual classification decisions. *Journal of Machine Learning Research* **11:** 1803-1831.
917 Beaumont, M. A., 2010 Approximate Bayesian computation in evolution and ecology. *Annual*
918 *review of ecology, evolution, and systematics* **41:** 379-406.
919 Begun, D. J., and C. F. Aquadro, 1992 Levels of naturally occurring DNA polymorphism
920 correlate with recombination rates in *D. melanogaster*. *Nature* **356:** 519-520.
921 Begun, D. J., A. K. Holloway, K. Stevens, L. W. Hillier, Y.-P. Poh *et al.*, 2007 Population
922 genomics: whole-genome analysis of polymorphism and divergence in Drosophila
923 simulans. *PLoS Biol.* **5:** e310.
924 Boitard, S., C. Schlötterer and A. Futschik, 2009 Detecting selective sweeps: a new approach
925 based on hidden Markov models. *Genetics* **181:** 1567-1578.
926 Brandvain, Y., A. M. Kenney, L. Flagel, G. Coop and A. L. Sweigart, 2014 Speciation and
927 introgression between Mimulus nasutus and Mimulus guttatus. *PLoS Genet.* **10:** e1004410.

Breiman, L., 2001 Statistical modeling: The two cultures (with comments and a rejoinder by the author). *Statistical science* **16:** 199-231.

Chan, A. H., P. A. Jenkins and Y. S. Song, 2012 Genome-wide fine-scale recombination rate variation in Drosophila melanogaster. *PLoS Genet.* **8:** e1003090.

Chan, J., V. Perrone, J. P. Spence, P. A. Jenkins, S. Mathieson *et al.*, 2018 A Likelihood-Free Inference Framework for Population Genetic Data using Exchangeable Neural Networks. *bioRxiv*.

Charlesworth, B., M. Morgan and D. Charlesworth, 1993 The effect of deleterious mutations on neutral molecular variation. *Genetics* **134:** 1289-1303.

Chilimbi, T. M., Y. Suzue, J. Apacible and K. Kalyanaraman, 2014 Project Adam: Building an Efficient and Scalable Deep Learning Training System, pp. 571-582 in *OSDI*.

Corbett-Detig, R., and R. Nielsen, 2017 A hidden Markov model approach for simultaneously estimating local ancestry and admixture time using next generation sequence data in samples of arbitrary ploidy. *PLoS Genet.* **13:** e1006529.

Elyashiv, E., S. Sattath, T. T. Hu, A. Strutsovsky, G. McVicker *et al.*, 2016 A genomic map of the effects of linked selection in Drosophila. *PLoS Genet.* **12:** e1006130.

Ewing, G., and J. Hermisson, 2010 MSMS: a coalescent simulation program including recombination, demographic structure and selection at a single locus. *Bioinformatics* **26:** 2064-2065.

Fay, J. C., and C.-I. Wu, 2000 Hitchhiking under positive Darwinian selection. *Genetics* **155:** 1405-1413.

Ferrer-Admetlla, A., M. Liang, T. Korneliussen and R. Nielsen, 2014 On detecting incomplete soft or hard selective sweeps using haplotype structure. *Mol. Biol. Evol.* **31:** 1275-1291.

Fontaine, M. C., J. B. Pease, A. Steele, R. M. Waterhouse, D. E. Neafsey *et al.*, 2015 Extensive introgression in a malaria vector species complex revealed by phylogenomics. *Science* **347:** 1258524.

Fu, Y.-X., and W.-H. Li, 1993 Statistical tests of neutrality of mutations. *Genetics* **133:** 693-709.

Gao, F., C. Ming, W. Hu and H. Li, 2016 New software for the fast estimation of population recombination rates (FastEPRR) in the genomic era. *G3: Genes, Genomes, Genetics* **6:** 1563-1571.

Garrigan, D., S. B. Kingan, A. J. Geneva, P. Andolfatto, A. G. Clark *et al.*, 2012 Genome sequencing reveals complex speciation in the Drosophila simulans clade. *Genome Res.* **22:** 1499-1511.

Garud, N. R., P. W. Messer, E. O. Buzbas and D. A. Petrov, 2015 Recent selective sweeps in North American Drosophila melanogaster show signatures of soft sweeps. *PLoS Genet.* **11:** e1005004.

Gazave, E., L. Ma, D. Chang, A. Coventry, F. Gao *et al.*, 2014 Neutral genomic regions refine models of recent rapid human population growth. *Proceedings of the National Academy of Sciences* **111:** 757-762.

Geneva, A. J., C. A. Muirhead, S. B. Kingan and D. Garrigan, 2015 A new method to scan genomes for introgression in a secondary contact model. *PLoS ONE* **10:** e0118621.

Gutenkunst, R. N., R. D. Hernandez, S. H. Williamson and C. D. Bustamante, 2009 Inferring the joint demographic history of multiple populations from multidimensional SNP frequency data. *PLoS Genet.* **5:** e1000695.

Hahn, M. W., 2008 Toward a selection theory of molecular evolution. *Evolution* **62:** 255-265.

Haller, B., and P. Messer, 2017 SLiM 2: Flexible, Interactive Forward Genetic Simulations. *Mol. Biol. Evol.* **34:** 230.

Hedrick, P. W., 2013 Adaptive introgression in animals: examples and comparison to new mutation and standing variation as sources of adaptive variation. *Mol. Ecol.* **22:** 4606-4618.

Hellenthal, G., G. B. Busby, G. Band, J. F. Wilson, C. Capelli *et al.*, 2014 A genetic atlas of human admixture history. *Science* **343:** 747-751.

Hermisson, J., and P. S. Pennings, 2005 Soft sweeps molecular population genetics of adaptation from standing genetic variation. *Genetics* **169:** 2335-2352.

Hey, J., 2009 Isolation with migration models for more than two populations. *Mol. Biol. Evol.* **27:** 905-920.

Hey, J., and J. Wakeley, 1997 A coalescent estimator of the population recombination rate. *Genetics* **145:** 833-846.

Hill, W. G., and A. Robertson, 1966 The effect of linkage on limits to artificial selection. *Genetics Research* **8:** 269-294.

Hornik, K., 1991 Approximation capabilities of multilayer feedforward networks. *Neural networks* **4:** 251-257.

Hudson, R. R., 1987 Estimating the recombination parameter of a finite population model without selection. *Genetics Research* **50:** 245-250.

Hudson, R. R., 2001 Two-locus sampling distributions and their application. *Genetics* **159:** 1805-1817.

Hudson, R. R., 2002 Generating samples under a Wright–Fisher neutral model of genetic variation. *Bioinformatics* **18:** 337-338.

Hudson, R. R., and N. L. Kaplan, 1985 Statistical properties of the number of recombination events in the history of a sample of DNA sequences. *Genetics* **111:** 147-164.

Jensen, J. D., Y. Kim, V. B. DuMont, C. F. Aquadro and C. D. Bustamante, 2005 Distinguishing between selective sweeps and demography using DNA polymorphism data. *Genetics* **170:** 1401-1410.

Joly, S., P. A. McLenachan and P. J. Lockhart, 2009 A statistical approach for distinguishing hybridization and incomplete lineage sorting. *The American Naturalist* **174:** E54-E70.

Jouppi, N. P., C. Young, N. Patil, D. Patterson, G. Agrawal *et al.*, 2017 In-datacenter performance analysis of a tensor processing unit, pp. 1-12 in *Proceedings of the 44th Annual International Symposium on Computer Architecture*. ACM.

Kaplan, N. L., R. Hudson and C. Langley, 1989 The" hitchhiking effect" revisited. *Genetics* **123:** 887-899.

Kelleher, J., A. M. Etheridge and G. McVean, 2016 Efficient coalescent simulation and genealogical analysis for large sample sizes. *PLoS Comput. Biol.* **12:** e1004842.

Kelleher, J., K. Thornton, J. Ashander and P. Ralph, 2018 Efficient pedigree recording for fast population genetics simulation. *bioRxiv***:** 248500.

Kelly, J. K., 1997 A test of neutrality based on interlocus associations. *Genetics* **146:** 1197-1206.

Kern, A. D., and D. R. Schrider, 2016 discoal: flexible coalescent simulations with selection. *Bioinformatics* **32:** btw556.

Kern, A. D., and D. R. Schrider, 2018 diploS/HIC: an updated approach to classifying selective sweeps. *bioRxiv***:** 267229.

Kim, Y., and R. Nielsen, 2004 Linkage disequilibrium as a signature of selective sweeps. *Genetics* **167:** 1513-1524.

Kim, Y., and W. Stephan, 2002 Detecting a local signature of genetic hitchhiking along a recombining chromosome. *Genetics* **160:** 765-777.

1021 Kingma, D. P., and J. Ba, 2014 Adam: A method for stochastic optimization. *arXiv preprint*
1022 *arXiv:1412.6980*.
1023 Kong, A., G. Thorleifsson, D. F. Gudbjartsson, G. Masson, A. Sigurdsson *et al.*, 2010 Fine-scale
1024 recombination rate differences between sexes, populations and individuals. *Nature* **467:**
1025 1099-1103.
1026 Korneliussen, T. S., A. Albrechtsen and R. Nielsen, 2014 ANGSD: analysis of next generation
1027 sequencing data. *BMC Bioinformatics* **15:** 356.
1028 Köster, U., T. Webb, X. Wang, M. Nassar, A. K. Bansal *et al.*, 2017 Flexpoint: An adaptive
1029 numerical format for efficient training of deep neural networks, pp. 1742-1752 in *Advances*
1030 *in Neural Information Processing Systems*.
1031 Krizhevsky, A., I. Sutskever and G. E. Hinton, 2012 Imagenet classification with deep
1032 convolutional neural networks, pp. 1097-1105 in *Advances in neural information processing*
1033 *systems*.
1034 Kulathinal, R. J., L. S. Stevison and M. A. Noor, 2009 The genomics of speciation in
1035 Drosophila: diversity, divergence, and introgression estimated using low-coverage
1036 genome sequencing. *PLoS Genet.* **5:** e1000550.
1037 Langley, C. H., K. Stevens, C. Cardeno, Y. C. G. Lee, D. R. Schrider *et al.*, 2012 Genomic
1038 variation in natural populations of *Drosophila melanogaster*. *Genetics* **192:** 533-598.
1039 Lawrence, S., C. L. Giles, A. C. Tsoi and A. D. Back, 1997 Face recognition: A convolutional
1040 neural-network approach. *IEEE transactions on neural networks* **8:** 98-113.
1041 Lawson, D. J., G. Hellenthal, S. Myers and D. Falush, 2012 Inference of population structure
1042 using dense haplotype data. *PLoS Genet.* **8:** e1002453.
1043 LeCun, Y., Y. Bengio and G. Hinton, 2015 Deep learning. *Nature* **521:** 436-444.
1044 LeCun, Y., L. Bottou, Y. Bengio and P. Haffner, 1998 Gradient-based learning applied to
1045 document recognition. *Proceedings of the IEEE* **86:** 2278-2324.
1046 Li, H., and R. Durbin, 2011 Inference of human population history from individual whole-
1047 genome sequences. *Nature* **475:** 493-496.
1048 Li, N., and M. Stephens, 2003 Modeling linkage disequilibrium and identifying recombination
1049 hotspots using single-nucleotide polymorphism data. *Genetics* **165:** 2213-2233.
1050 Lin, K., A. Futschik and H. Li, 2013 A fast estimate for the population recombination rate based
1051 on regression. *Genetics* **194:** 473-484.
1052 Lin, K., H. Li, C. Schlötterer and A. Futschik, 2011 Distinguishing positive selection from
1053 neutral evolution: boosting the performance of summary statistics. *Genetics* **187:** 229-244.
1054 Lipson, M., P.-R. Loh, A. Levin, D. Reich, N. Patterson *et al.*, 2013 Efficient moment-based
1055 inference of admixture parameters and sources of gene flow. *Mol. Biol. Evol.* **30:** 1788-
1056 1802.
1057 Liu, X., and Y.-X. Fu, 2015 Exploring population size changes using SNP frequency spectra.
1058 *Nat. Genet.* **47:** 555-559.
1059 Loh, P.-R., M. Lipson, N. Patterson, P. Moorjani, J. K. Pickrell *et al.*, 2013 Inferring admixture
1060 histories of human populations using linkage disequilibrium. *Genetics* **193:** 1233-1254.
1061 Lu, X., Z. Lin, X. Shen, R. Mech and J. Z. Wang, 2015 Deep multi-patch aggregation network
1062 for image style, aesthetics, and quality estimation, pp. 990-998 in *Proceedings of the IEEE*
1063 *International Conference on Computer Vision*.
1064 Marth, G. T., E. Czabarka, J. Murvai and S. T. Sherry, 2004 The allele frequency spectrum in
1065 genome-wide human variation data reveals signals of differential demographic history in
1066 three large world populations. *Genetics* **166:** 351-372.

Martin, S. H., K. K. Dasmahapatra, N. J. Nadeau, C. Salazar, J. R. Walters *et al.*, 2013 Genome-wide evidence for speciation with gene flow in Heliconius butterflies. *Genome Res.* **23:** 1817-1828.

Maynard Smith, J., and J. Haigh, 1974 The hitch-hiking effect of a favourable gene. *Genet. Res.* **23:** 23-35.

McVean, G. A., S. R. Myers, S. Hunt, P. Deloukas, D. R. Bentley *et al.*, 2004 The fine-scale structure of recombination rate variation in the human genome. *Science* **304:** 581-584.

Mitchell, T. M., 1997 Artificial neural networks. *Machine Learning* **45:** 81-127.

Nair, V., and G. E. Hinton, 2010 Rectified linear units improve restricted boltzmann machines, pp. 807-814 in *Proceedings of the 27th international conference on machine learning (ICML-10)*.

Nei, M., and W.-H. Li, 1979 Mathematical model for studying genetic variation in terms of restriction endonucleases. *Proceedings of the National Academy of Sciences* **76:** 5269-5273.

Nielsen, R., and J. Wakeley, 2001 Distinguishing migration from isolation: a Markov chain Monte Carlo approach. *Genetics* **158:** 885-896.

Nielsen, R., S. Williamson, Y. Kim, M. J. Hubisz, A. G. Clark *et al.*, 2005 Genomic scans for selective sweeps using SNP data. *Genome Res.* **15:** 1566-1575.

Pavlidis, P., J. D. Jensen and W. Stephan, 2010 Searching for footprints of positive selection in whole-genome SNP data from nonequilibrium populations. *Genetics* **185:** 907-922.

Price, A. L., A. Tandon, N. Patterson, K. C. Barnes, N. Rafaels *et al.*, 2009 Sensitive detection of chromosomal segments of distinct ancestry in admixed populations. *PLoS Genet.* **5:** e1000519.

Pudlo, P., J.-M. Marin, A. Estoup, J.-M. Cornuet, M. Gautier *et al.*, 2016 Reliable ABC model choice via random forests. *Bioinformatics* **32:** 859-866.

Pybus, M., P. Luisi, G. M. Dall'Olio, M. Uzkudun, H. Laayouni *et al.*, 2015 Hierarchical boosting: a machine-learning framework to detect and classify hard selective sweeps in human populations. *Bioinformatics* **31:** 3946-3952.

Racimo, F., D. Marnetto and E. Huerta-Sanchez, 2016 Signatures of archaic adaptive introgression in present-day human populations. *Mol. Biol. Evol.* **34:** 296-317.

Rasmussen, M. D., M. J. Hubisz, I. Gronau and A. Siepel, 2014 Genome-wide inference of ancestral recombination graphs.

Ribeiro, M. T., S. Singh and C. Guestrin, 2016 Why should i trust you?: Explaining the predictions of any classifier, pp. 1135-1144 in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM.

Ronen, R., N. Udpa, E. Halperin and V. Bafna, 2013 Learning natural selection from the site frequency spectrum. *Genetics* **195:** 181-193.

Rosenzweig, B. K., J. B. Pease, N. J. Besansky and M. W. Hahn, 2016 Powerful methods for detecting introgressed regions from population genomic data. *Mol. Ecol.* **25:** 2387-2397.

Rumelhart, D. E., G. E. Hinton and R. J. Williams, 1986 Learning representations by back-propagating errors. *Nature* **323:** 533.

Sankararaman, S., S. Mallick, M. Dannemann, K. Prüfer, J. Kelso *et al.*, 2014 The genomic landscape of Neanderthal ancestry in present-day humans. *Nature* **507:** 354-357.

Schiffels, S., and R. Durbin, 2014 Inferring human population size and separation history from multiple genome sequences. *Nat. Genet.* **46:** 919-925.

Schlötterer, C., R. Tobler, R. Kofler and V. Nolte, 2014 Sequencing pools of individuals—mining genome-wide polymorphism data without big funding. *Nature Reviews Genetics* **15:** 749.

Schrider, D., J. Ayroles, D. R. Matute and A. D. Kern, 2018 Supervised machine learning reveals introgressed loci in the genomes of *Drosophila simulans* and *D. sechellia*. *PLoS Genet.* **14:** e1007341.

Schrider, D. R., and A. D. Kern, 2016 S/HIC: Robust Identification of Soft and Hard Sweeps Using Machine Learning. *PLoS Genet.* **12:** e1005928.

Schrider, D. R., and A. D. Kern, 2017 Soft sweeps are the dominant mode of adaptation in the human genome. *Mol. Biol. Evol.* **34:** 1863–1877.

Schrider, D. R., and A. D. Kern, 2018 Supervised Machine Learning for Population Genetics: A New Paradigm. *Trends Genet.* **34:** 301-312.

Schrider, D. R., F. K. Mendes, M. W. Hahn and A. D. Kern, 2015 Soft shoulders ahead: spurious signatures of soft and partial selective sweeps result from linked hard sweeps. *Genetics* **200:** 267-284.

Sheehan, S., and Y. S. Song, 2016 Deep learning for population genetic inference. *PLoS Comput. Biol.* **12:** e1004845.

Simonsen, K. L., G. A. Churchill and C. F. Aquadro, 1995 Properties of statistical tests of neutrality for DNA polymorphism data. *Genetics* **141:** 413-429.

Simonyan, K., and A. Zisserman, 2014 Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.

Smith, J., G. Coop, M. Stephens and J. Novembre, 2018 Estimating time to the common ancestor for a beneficial allele. *Mol. Biol. Evol.*

Sohn, K.-A., Z. Ghahramani and E. P. Xing, 2012 Robust estimation of local genetic ancestry in admixed populations using a nonparametric Bayesian approach. *Genetics* **191:** 1295-1308.

Srivastava, N., G. Hinton, A. Krizhevsky, I. Sutskever and R. Salakhutdinov, 2014 Dropout: A simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research* **15:** 1929-1958.

Sugden, L. A., E. G. Atkinson, A. P. Fischer, S. Rong, B. M. Henn *et al.*, 2018 Localization of adaptive variants in human genomes using averaged one-dependence estimation. *Nature Communications* **9:** 703.

Szegedy, C., W. Liu, Y. Jia, P. Sermanet, S. Reed *et al.*, 2015 Going deeper with convolutions, pp. in *CVPR*.

Tajima, F., 1989 Statistical method for testing the neutral mutation hypothesis by DNA polymorphism. *Genetics* **123:** 585-595.

Tennessen, J. A., A. W. Bigham, T. D. O'Connor, W. Fu, E. E. Kenny *et al.*, 2012 Evolution and functional impact of rare coding variation from deep sequencing of human exomes. *Science* **337:** 64-69.

Teshima, K. M., and H. Innan, 2009 mbs: modifying Hudson's ms software to generate samples of DNA sequences with a biallelic site under selection. *BMC Bioinformatics* **10:** 166.

Thornton, K. R., 2014 A C++ template library for efficient forward-time population genetic simulation of large populations. *Genetics* **198:** 157-166.

Turner, T. L., M. W. Hahn and S. V. Nuzhdin, 2005 Genomic islands of speciation in Anopheles gambiae. *PLoS Biol.* **3:** e285.

Voight, B. F., S. Kudaravalli, X. Wen and J. K. Pritchard, 2006 A map of recent positive selection in the human genome. *PLoS Biol.* **4:** e72.

Vy, H. M. T., and Y. Kim, 2015 A Composite-Likelihood Method for Detecting Incomplete Selective Sweep from Population Genomic Data. *Genetics* **200:** 633-649.

Watterson, G., 1975 On the number of segregating sites in genetical models without recombination. *Theor. Popul. Biol.* **7:** 256-276.

Yu, F., and V. Koltun, 2015 Multi-scale context aggregation by dilated convolutions. *arXiv preprint arXiv:1511.07122*.

Zaheer, M., S. Kottur, S. Ravanbakhsh, B. Poczos, R. R. Salakhutdinov *et al.*, 2017 Deep sets, pp. 3394-3404 in *Advances in Neural Information Processing Systems*.

SUPPLEMENTARY TABLE LEGENDS

Table S1: Demographic parameter ranges used to simulate 3-epoch population size histories.

Table S2: The effect of different neural network input/output/architecture hyperparameters on demographic inference error.