Check for updates

Received: 30 June 2020 | Revised: 26 January 2021 | Accepted: 5 February 2021

DOI: 10.1111/1755-0998.13355

SPECIAL ISSUE

MOLECULAR ECOLOGY RESOURCES WILEY

Chromosome-scale inference of hybrid speciation and admixture with convolutional neural networks

Paul D. Blischak[1,2] | Michael S. Barker[1] | Ryan N. Gutenkunst[2]

[1]Department of Ecology and Evolutionary Biology, University of Arizona, Tucson, AZ, USA

[2]Department of Molecular and Cellular Biology, University of Arizona, Tucson, AZ, USA

Correspondence
Paul D. Blischak, Department of Ecology and Evolutionary Biology, University of Arizona, Tucson, AZ 85721, USA.
Email: pblischak@arizona.edu; paul.blischak@gmail.com

Funding information
National Institute of General Medical Sciences, Grant/Award Number: R01GM127348; National Science Foundation, Grant/Award Number: IOS-1811784

Abstract

Inferring the frequency and mode of hybridization among closely related organisms is an important step for understanding the process of speciation and can help to uncover reticulated patterns of phylogeny more generally. Phylogenomic methods to test for the presence of hybridization come in many varieties and typically operate by leveraging expected patterns of genealogical discordance in the absence of hybridization. An important assumption made by these tests is that the data (genes or SNPs) are independent given the species tree. However, when the data are closely linked, it is especially important to consider their nonindependence. Recently, deep learning techniques such as convolutional neural networks (CNNs) have been used to perform population genetic inferences with linked SNPs coded as binary images. Here, we use CNNs for selecting among candidate hybridization scenarios using the tree topology $(((P_1, P_2), P_3), \text{Out})$ and a matrix of pairwise nucleotide divergence (d_{XY}) calculated in windows across the genome. Using coalescent simulations to train and independently test a neural network showed that our method, HyDe-CNN, was able to accurately perform model selection for hybridization scenarios across a wide breath of parameter space. We then used HyDe-CNN to test models of admixture in *Heliconius* butterflies, as well as comparing it to phylogeny-based introgression statistics. Given the flexibility of our approach, the dropping cost of long-read sequencing and the continued improvement of CNN architectures, we anticipate that inferences of hybridization using deep learning methods like ours will help researchers to better understand patterns of admixture in their study organisms.

KEYWORDS
admixture, convolutional neural networks, deep learning, gene flow, hybridization, model selection

1 | INTRODUCTION

Hybridization is a widely observed phenomenon in nature that continues to be appreciated across diverse lineages in the Tree of Life. From some of the earliest systematic investigations of hybridization and introgression in irises (Anderson, 1949; Anderson & Stebbins, 1954) to more recent studies of selection on introgressed loci from Neanderthals into non-African humans (Chen et al. 2020; Harris & Nielsen, 2016; Juric et al. 2016), understanding the importance of genetic exchanges between otherwise isolated lineages is of great interest to researchers across many fields in evolutionary genetics. Nevertheless, many studies aimed at

 © 2021 John Wiley & Sons Ltd

inferring patterns of hybridization, especially in nonmodel taxa, lack the resolution to determine the specific mode of hybridization occurring within their study organism(s) (i.e. homoploid hybrid speciation versus a pulse of admixture after divergence) owing to limited knowledge about haplotype structure and built-in methodological assumptions regarding independence among SNP. However, the dropping cost of long-read sequencing technologies such as Oxford Nanopore and Pacific Biosciences is quickly democratizing the acquisition of chromosome-scale reference genomes for nonmodel species (Amarasinghe et al. 2020). As more laboratories obtain these references, it will be especially important to explore methods of hybridization inference that can leverage the information encoded by the contiguous variants along the genome.

In the absence of a genome reference, myriad summary statistics have been described for the detection of introgression and hybrid speciation from unlinked, biallelic SNPs. Among the earliest of these statistics is *D* (Green et al. 2010), or the 'ABBA-BABA' statistic, which is used to test for a statistically significant excess of ABBA or BABA site patterns for biallelic sites evolving along genealogies from the species tree (((*P*$_1$, *P*$_2$), *P*$_3$), Out). In the absence of introgression, we expect only incomplete lineage sorting (ILS) to generate ABBA or BABA site patterns, a result that derives from the coalescent model (Kingman, 1982). Deviations from the coalescent expectations of site patterns evolving along genealogies form the basis of nearly all tests for introgression, including tests on five-taxon trees (Eaton & Ree, 2013; Pease & Hahn, 2015), tests on *N*-taxon trees (Elworth et al., 2018), tests that assume a specific mode of hybridization (e.g. hybrid speciation; Blischak et al. 2018; Kubatko & Chifman, 2019) and more recent approaches that incorporate sequence divergence into their test statistics (Forsythe et al. 2020; Hahn & Hibbins, 2019; Hibbins & Hahn, 2019). And while some of these statistics aim to distinguish among modes of hybridization and admixture or the direction of introgression, they all assume that genealogies across sites or genes are unlinked. The shared genealogical information contained in linked SNPs, especially about the size and coalescence time of linkage blocks, is an important feature of chromosome-scale data that is ignored in many inferences of hybridization, but which can likely provide greater power for distinguishing among more complex models.

Incorporating linkage into inferences of hybridization requires modelling both the coalescent process and the history of recombination along a chromosome, generating a highly complex object called the ancestral recombination graph (ARG; Griffiths & Marjoram, 1997; Hudson, 1983). However, because the complexity of the ARG typically makes it unwieldy to work with and nearly impossible to estimate, researchers developed an approximation called the sequentially Markovian coalescent (SMC; McVean & Cardin, 2005), which greatly reduces the headaches associated with ARGs by restricting the space of possible coalescent events for marginal genealogies along a chromosome, creating an approximation that is both accurate and computationally tractable (but see methods Relate and tsinfer below). Applications of the SMC model have primarily come in the form of coalescent hidden Markov models (HMMs), which model genealogies or coalescent times (hidden states) of variants along the genome (observed states) such that adjacent variants either share a hidden state (no recombination) or are in different hidden states because of a recombination event (Dutheil et al. 2009; Hobolth et al. 2007). Some of the most well-known uses of coalescent HMMs are for the inference of pairwise coalescent times and population size trajectories. This includes methods that require phased genomes such as PSMC (Li & Durbin, 2011) and MSMC (Schiffels & Durbin, 2014), as well as methods that do not require known phase among the sampled individuals such as SMC++ (Terhorst et al. 2017). Two recent approaches, Relate (Speidel et al. 2019) and tsinfer (Kelleher et al. 2019), have even managed to use HMMs for approximating the full ARG by inferring the genealogical relationships among thousands of samples from entire chromosomes. Phylogenetic applications of coalescent HMMs include the original CoalHMM implementation for inferring parameters among human, chimp and gorilla (Dutheil et al. 2009; Hobolth et al. 2007), although these approaches use a much simpler representation of the ARG that does not require the SMC model. Phylogenetic HMMs for the inference of admixture and hybridization have also been developed and include approaches such as PhyloNet-HMM (Liu et al. 2014) and updated version of the CoalHMM framework to model gene flow (Mailund et al. 2012).

The ability of coalescent HMMs and the SMC to infer complicated population histories comes from leveraging the correlation between genealogies along the chromosome resulting from linkage. However, as models become more complex, trying to incorporate and estimate all relevant parameters in a likelihood-based framework can become computationally prohibitive. Fortunately, efficient tools for simulating genome-scale data (e.g. msprime and SLiM 3; Haller & Messer, 2019; Kelleher et al. 2016) provide opportunities to explore likelihood-free inference methods such as approximate Bayesian computation (ABC; Beaumont et al. 2002) and supervised machine learning (Schrider & Kern, 2018; Sheehan & Song, 2016). One particular type of supervised machine learning that has received increasing attention in population genomics and phylogenetics is deep learning, or the use of neural networks for model selection and/or parameter inference (LeCun et al. 2015). One type of deep learning algorithm that has recently been used to account for linkage among variant sites for different inference problems in population genomics is convolutional neural networks (CNNs; Flagel et al. 2018). Convolutional neural networks are typically employed for image recognition tasks because they focus on modelling structure in images by allowing adjacent pixels to share parameters in the network (LeCun et al. 1998). In a population genomic or phylogenetic setting, the input image for a CNN is simply any representation of genetic variation along the chromosome (e.g. a genotype matrix), typically among samples from a population or across species (Flagel et al. 2018; Suvorov et al. 2019). Because sites that are close to one another are more likely to come from the same ancestral recombination block, patterns of variation will reflect this correlation, allowing a CNN to be trained on genome-wide patterns of variation using simulations.

Here, we explore the use of CNNs to perform model selection for different modes of hybridization using chromosome-scale representations of genomic data among pairs of species. Our approach uses simulations to train a CNN with pairwise sequence divergence calculated in windows across the chromosome, which captures the correlation generated by linkage among (unphased) variant sites. Training, validation and testing on independent data allowed us to evaluate the prediction accuracy of our trained CNNs to distinguish among phylogenetic models of hybridization and admixture. We then compared the prediction accuracy of the trained CNNs to a set of more traditional, phylogenetic summary statistics to see if accounting for linkage with the CNN resulted in better accuracy for model selection. Finally, we used our CNN approach to test among different hypotheses for the mode of admixture between populations of *Heliconius* butterflies. Overall, we show that image-based representations of chromosome-scale, phylogenomic data lend themselves well to predicting phylogenetic models of hybridization. As genome reference sequences continue to become available, we anticipate that approaches like ours will be increasingly useful not only for inferring hybridization but also for testing other hypotheses about evolutionary patterns as well.

2 | MATERIALS AND METHODS

2.1 | Representing phylogenomic data as an image

Using a similar conceptual set-up as coalescent HMMs, we want to represent phylogenomic data that encodes both coalescence times along the chromosome and phylogenetic relationships among the sampled species. These two axes, chromosomal and phylogenetic, can be thought of as structuring genomic data by encoding linkage among variant sites along the chromosome and common ancestry across the phylogeny, such that a two-dimensional representation can be treated as an image used to train a CNN (see example image in Figure 1A). For the phylogenetic axis, we consider the four-taxon phylogeny typically used for tests of introgression, $(((P_1, P_2), P_3), \text{Out})$, and represent it by considering all population pairs ordered by increasing phylogenetic distance: $P_1 \times P_2$, $P_1 \times P_3$, $P_2 \times P_3$, $P_1 \times \text{Out}$, $P_2 \times \text{Out}$, $P_3 \times \text{Out}$. For the chromosomal axis, we first divide the chromosome into W windows. Next, within each window, we calculate the average number of nucleotide differences for all sampled chromosomes between each pair of species (Nei's genetic distance; $d_{XY,w}$) for species pairs $X \times Y$ and window w $w \in \{1, \ldots, W\}$ (Nei, 1987; Nei & Li, 1979). If there are n_X and n_Y chromosomes from species X and Y, respectively, this gives us a distribution of $n_X \times n_Y$ values of $d_{XY,w}$ within each window, which we treat as a proxy for the distribution of pairwise coalescent times between chromosomes sampled from each species. The relationship between coalescence times and genetic variation under the infinite sites model is a classical result in population genetics (Tajima, 1983; Watterson, 1975), and although other statistics for measuring genetic differences between populations exist (e.g. F_{ST}; Wright, 1931, 1943), we use d_{XY} here because it is a simple measure of divergence between chromosomes.

2.2 | CNN architecture

The set-up for our neural network is based on the LeNet architecture, which was introduced as one of the earliest uses of CNNs for the identification of hand-written characters (LeCun et al. 1998). In our implementation (Figure 1B), we start by repeating a sequence of three layers r times: a two-dimensional convolution layer with x_s filters (s = 1,...r), a two-dimensional average pooling layer and a dropout layer. In our 2D convolution layer, a 4 × 2 filter walks across the input image, stepping across pixels in the rows and columns one at a time, and produces a single numerical output that is the convolution (dot product) of the filter values and the image values at each step.

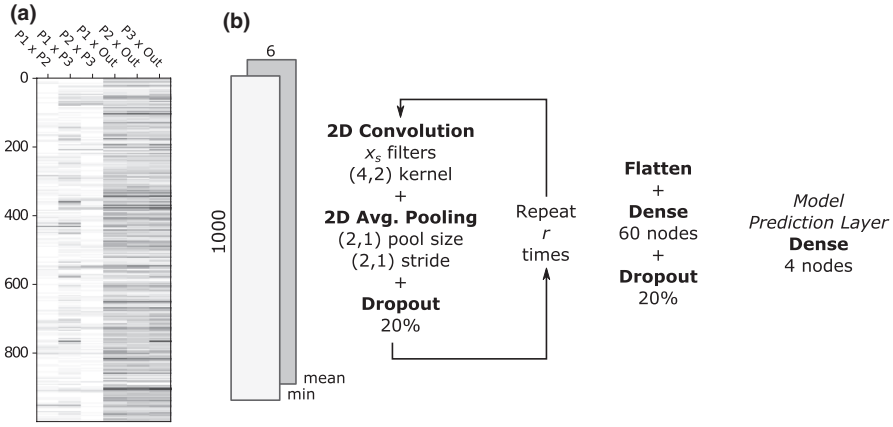

FIGURE 1 Neural network architecture for HyDe-CNN. (A) A sample image of minimum d_{XY} across 1000 windows (rows) for the six pairwise comparisons among the species (columns). In this image, P_2 is a hybrid species between P_1 and P_3, which is illustrated by the lower (lighter) values of d_{XY} for $P_1 \times P_2$ and $P_2 \times P_3$ compared to $P_1 \times P_3$. (B) A simplified representation of the convolutional neural network architecture of HyDe-CNN. A breakdown of how the CNN processes the input images through each layer of the network is provided in the Supplemental Materials (Table S1)

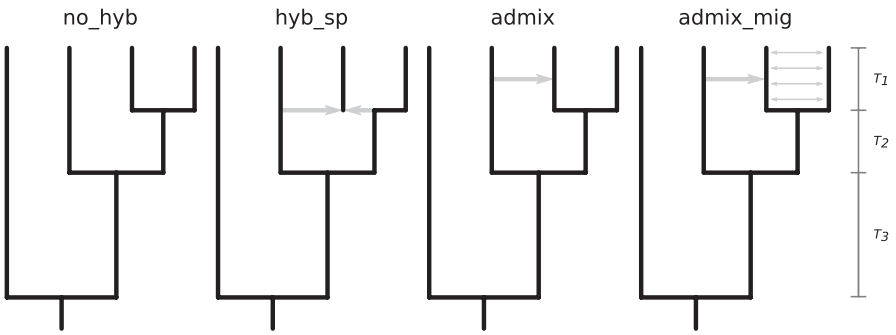

FIGURE 2 Models used to simulate genomic data for hybridization detection. From left to right, they represent (1) no hybridization, (2) hybrid speciation, (3) pulsed admixture, and (4) pulsed admixture with ongoing gene flow. Parameters for all models are drawn from distributions as described in the Materials and Methods

The values in the filter are randomly initialized such that the feature extraction from the underlying image data is stochastic. However, when the CNN weights are optimized during training, extracted features that lead to higher prediction accuracy will be given higher weight. Similarly, the 2D average pooling layer returns the mean of a 2 × 1 block of pixels, stepping down two pixels as it walks across the image. This step serves to combine the information in adjacent pixels by taking their average, and the step size of two pixels also reduces the number of rows in the original input by half. The dropout layer that follows these layers randomly excludes a portion of the input and helps with preventing overfitting to the training data. The number of times this sequence of layers is repeated determines how reduced the original input image will be and therefore also determines the number of parameters that the network needs to estimate. For our network, we chose to repeat this sequence four times using 12, 24, 36 and 48 as the number of filters in each of the four cycles. The next layer is a dense or fully connected layer with 60 nodes, followed by the final four-node dense layer that produces the model prediction. All layers except for the final layer use rectified linear unit activation functions (ReLU), which are constructed to try and capture nonlinear interactions in the input image (Agarap, 2018). The final layer uses the softmax activation function, which produces output weights, p_i ($\sum_i p_i = 1$), that are proportional to the support given to each model by the network. Network parameters are then estimated using a categorical cross-entropy loss function and Adam optimizer (Kingma & Ba, 2017). We refer to our architecture for hybridization inference as HyDe-CNN, which we specified with TensorFlow version 2.1.0 using the tf.keras interface version 2.2.4-tf (Abadi et al., 2016). A breakdown of how the shape of the input image changes as it is processed by each layer of the CNN can be found in Table S1.

2.3 | Simulating training, validation and test data

To test the ability of our data representation for distinguishing among models of hybridization with HyDe-CNN, we simulated chromosome-scale data under four models, depicted in Figure 2: no hybridization (1; *no_hyb*), hybrid speciation (2; *hyb_sp*), admixture (3; *admix*) and admixture with migration (4; *admix_mig*). All simulations were conducted in Python v3.7.6 with msprime v0.7.4 msprime using parameters drawn from the following distributions (numbers indicate which models use each parameter):

- [1–4] Sequence length (L): Discrete Uniform(1×10^7, 5×10^7, step $=1 \times 10^5$).
- [1–4] Mutation rate (μ) and recombination rate (r): Uniform(2.5×10^{-9}, 2.5×10^{-8}).
- [1–4] Divergence times (T_1, T_2, T_3): Gamma(10.0, 0.1), Gamma(20.0, 0.1), Gamma(40.0, 0.1).
- [3,4] Admixture time (T_{mix}): Uniform($0.1 \times T_1$, $0.9 \times T_1$).
- [2] Hybridization fraction (γ): Uniform(0.25, 0.75).
- [3,4] Admixture proportion (f): Uniform(0.01, 0.25).
- [4] Migration rate (m): Uniform(2.5×10^{-4}, 5×10^{-4}).

Parameterizing the models in this way allowed us to explore a wide range of scenarios including variation in the timing and proportion of hybridization and admixture, variable mutation and recombination rates, and variable levels of sequence diversity ($\theta = 4N_A L\mu$ ranges from 100 to 5000; N_A = 1000). The resulting tree sequences from each msprime simulation run were summarized by calculating the number of pairwise differences between chromosomes for all pairs of species using 1000 equally spaced windows with the diversity() function in tskit version 0.2.3 (Kelleher et al. 2018; Ralph et al. 2020). For each species, we sampled five chromosomes, leading to 25 pairwise values of $d_{XY,w}$ within each window for each pair of species. To summarize this distribution of pairwise $d_{XY,w}$, we calculated the minimum and mean values, storing these in numpy arrays for downstream CNN training, validation and testing using the minimum alone, the mean alone and the minimum and mean as two channels of the input image (Van DerWalt et al. 2011).

Next, we conducted 20,000 simulations for each model at three different divergence scalings in coalescent units, 0.5 CUs (high ILS), 1.0 CUs (medium ILS) and 2.0 CUs (low ILS), for a total of 240,000 simulated data sets. In total, we trained nine separate CNNs, one for each input type (minimum, mean and minimum + mean) and divergence scaling combination. For each CNN, we split the simulated data into training, validation and test data using 15,000 images from each model for training and 2500 images for both validation and testing. The input order of the images was shuffled so that images from different models were processed randomly by the CNN. Each individual image was also normalized by its maximum value to produce inputs with values between 0 and 1. Training and validation data were then fed to the CNN to fit the network parameters with a default learning rate of 0.001. We used the EarlyStopping callback within TensorFlow to stop training by monitoring the validation loss

(CNN categorical cross-entropy score on validation data) rather than setting a certain number of training iterations, which helps to prevent overfitting to the training data. After training was completed, we ran the independent test samples through the final CNN to perform model selection. The model selection results were then processed with functions in scikit-learn version 0.22.2 (Pedregosa et al. 2011) in Python to calculate prediction accuracy metrics and to generate confusion matrices.

To assess an alternative architecture to HyDe-CNN, we implemented a modified version of the network used by Flagel et al. (2018) for inferring introgression to train and test on our simulated input images. The biggest difference between HyDe-CNN and the network used by Flagel et al. is that it uses one-dimensional convolutional and average pooling layers, meaning that all pairs of species are used in these layers at once, rather than two at a time as in our architecture. Using one-dimensional layers in the network also limited us to only test this architecture on the minimum and mean d_{XY} images since one-dimensional layers in TensorFlow cannot accept input data with more than one channel due to restrictions on the expected input dimensions. All parameters in our implementation of the modified Flagel et al. architecture were kept the same except for the number of filters, which we reduced from 256 for the first layer and 128 for all other layers to 64 and 32, respectively. The primary reason for reducing the size of the network was because we trained all of our CNNs on a MacBook Pro laptop (2.3 GHz Intel Core i9, 16 GB RAM, 8 cores), which does not have a graphics processing unit (GPU) capable of training a network of the original size specified by Flagel et al. All other training steps were completed exactly as above for the HyDe-CNN architecture.

2.4 | Assessing model accuracy

To further explore the predictions of the trained CNN, we conducted a second set of 10,000 simulations for each model and divergence scaling combination to understand whether prediction accuracy is associated with the actual biological parameters being used for the simulations. Data were simulated and processed into the formatted images of pairwise nucleotide divergence using the same methods described above. We focused here on the HyDe-CNN model trained using minimum pairwise divergence and processed all data sets in Python with TensorFlow. We then recorded the true model and parameter values used to simulate each data set, as well as the predicted model and its weight. Results were then plotted with GGPLOT2 version 3.2.1 (Wickham, 2009) in R v3.6.1 (R Core Team, 2019) to visualize the relationship between the simulation parameters and model predictions.

2.5 | Comparison with summary statistics

For each of the 10,000 simulations from each model at the different divergence scaling values used in the previous subsection, we also calculated a set of summary statistics constructed to detect introgression from biallelic SNP frequencies on four-taxon phylogenies. We chose three statistics, D (Green et al. 2010), f_{hom} (Durand et al. 2011) and D_p (Hamlin et al. 2020), which are similar in form and are frequently used in phylogenetic studies to detect patterns of hybridization. Rather than analysing each statistic separately, we used the calculated values of the three statistics as predictor variables in a random forest classifier to jointly consider their ability to identify models of hybridization. To train a random forest classifier for each divergence scaling, we split the simulated data sets from each model into 7500 training and 2500 testing samples and used the R packages ABCRF version 1.8.1 (Pudlo et al. 2016) and CARET verson 6.0-86 (Kuhn, 2008). Despite our use of the ABCRF package, it should be noted that this comparison is not intended to be a fully implemented ABC algorithm but simply uses the package as a wrapper to implement the random forest classifier. Models were trained using the abcrf() function with 1000 trees to build the random forest classifier and default values for all other options. The trained classifier was then used to predict the class of the testing samples, and error rates were summarized using confusion matrices.

2.6 | Empirical example: Heliconius butterflies

Butterflies from the genus *Heliconius* have been studied extensively as a model system for non-bifurcating patterns of phylogenetic descent and the interplay of introgression and phenotypic evolution (Dasmahapatra et al. 2012; Edelman et al. 2019; Moest et al. 2020; Nadeau et al. 2013). Occurring primarily in Central and South America, *Heliconius* butterflies have conspicuous wing coloration patterning and are perhaps most famous for being one of the systems studied by Henry Walter Bates when describing his eponymous form of mimicry (Bates, 1862). Previous work to uncover the extent of genetic exchanges among *Heliconius* species has quantified and described variation in the amount of admixture between sympatric populations of different species (Martin et al. 2013, 2019). Here, we sought to leverage this previous knowledge to further dissect these patterns of admixture using deep learning.

To test different models of admixture in *Heliconius*, we downloaded variant calls from Dryad for four taxa: *H. cydno* (*cyd*; *H. c. chioneus* + *H. c. zelinde*), *H. melpomene*-West (*mel*-W; *H. m. rosina* + *H. m. vulcanus*), *H. melpomene*-East (*mel*-E *H. m. malleti* + *H. m. amaryllis*) and *H. numata* (*num*) (https://doi.org/10.5061/dryad.sk2pd88; Martin et al. 2019). Previous work has shown that admixture occurs between sympatric *cyd* and *mel*-W west of the Andes mountains, typically going from *cyd* into *mel*-W (although some bidirectional introgression likely occured; Martin et al. 2019). We therefore wanted to test whether we could distinguish between two alternative patterns of admixture between these two populations, (1) a single pulse of ancient admixture or (2) continuous gene flow at low frequencies, as well as including

a model with no hybridization as a null hypothesis. To compare these models, we simulated data with msprime for 9.5 Mb of *Heliconius melpomene* chromosome five (positions 200,000 to 9,700,000) using a previously estimated recombination map for *Heliconius* (Davey et al. 2017) and the demographic model used for simulations in Martin et al. (2019). The fraction of admixture, *f*, was simulated using a random uniform distribution on the interval [0.3,0.4], consistent with levels observed in these taxa (Martin et al. 2013, 2019). The divergence time between *mel*-E and *mel*-W was 0.5 CU in the past, followed by the divergence between these two taxa and *cyd* at 1.5 CU, and finally the divergence of the outgroup, *num*, at 4.0 CU. To make the simulations more computationally feasible, the simulations were scaled to a population size of 2000 individuals with a mutation rate of $\mu = 7.7 \times 10^{-7}$. This mutation rate was chosen because it produced approximately the same number of segregating sites in the simulated data compared to the real data. For the ancient admixture model, we simulated the timing of admixture from a uniform distribution from [0.25,0.5] CU in the past. For the continuous gene flow model, we divided the simulated value of *f* by the number of generations (0.5 CU × 2 × 2000 = 1000 generations) to give a cumulative level of admixture equal to *f* over the interval since the divergence of *mel*-W and *mel*-E.

We then simulated 20,000 data sets from each of the admixture models, plus the no hybridization scenario, using msprime and tskit as before to calculate d_{XY} among the population pairs in windows across the chromosome. We sampled 10 chromosomes for the populations representing *mel*-E, *mel*-W and *cyd*, and sampled four chromosomes from the outgroup *num* population, dividing the 9.5 Mb of simulated sequence into 950 equally sized, 10-kb windows. The distribution of pairwise d_{XY} values for each pair of populations was again calculated using the diversity() function and was summarized using the minimum, mean and minimum+mean to generate three different types of input images for training. All aspects of training remained the same as for our original set of exploratory simulations, except for only having three models to choose from and using only 950 windows.

To test the predictions of the trained CNN for *Heliconius*, we used the pysam library (version 0.15.3; https://github.com/pysam-developers/pysam) in Python to parse and process variant calls in VCF format on chromosome five using only positions 200,000 to 9,700,000 (Li et al., 2009). A total of 20 individuals (40 chromosomes) were sampled from each of *mel*-W, *mel*-E and *cyd*, so instead of using all samples, we randomly drew five individuals (10 chromosomes) from each population to match our simulations. Only two individuals of *num* were available, so we used both in all calculations. We repeated this random sampling of individuals 100 times, calculating pairwise values of d_{XY}, summarizing the distribution using the minimum, mean and minimum + mean, and running the resulting images through the corresponding trained CNN. We then assessed support for each of the models across the 100 replicates by recording how frequently each model received the highest weight.

TABLE 1 Overall prediction accuracy of the different machine learning algorithms used for model selection on independent test data. Divergence scaling is given in coalescent units (CU)

Method	Input	0.5	1.0	2.0
HyDe-CNN	Min d_{XY}	0.833	0.919	0.944
	Mean d_{XY}	0.716	0.864	0.924
	Min +Mean d_{XY}	0.845	0.910	0.940
Flagel et al.	Min d_{XY}	0.766	0.884	0.902
	Mean d_{XY}	0.536	0.774	0.876
Random Forest	D, f_{hom}, D_p	0.627	0.762	0.867

3 | RESULTS

3.1 | Performance of trained CNNs

Training HyDe-CNN and the modified Flagel et al. network was not overly time-consuming, usually taking between 5 and 10 minutes. It also usually only took a few training epochs for the validation loss score to plateau, suggesting that the networks quickly learn the information contained within the input images. This may also be because our networks are quite small compared to other more intensive implementations of deep neural networks (e.g. DeepVariant; Poplin et al. 2018), a point we return to in the **Discussion**. Prediction with the trained networks was only constrained by the amount of time it took to load the network into memory, making this step take just a few seconds. Not surprisingly, the most time intensive step for training the CNNs was generating the simulated input data images. Simulating 20,000 images from a single model usually took ~24 h but depended on the total number of generations needing to be simulated. However, because each simulated image coming from a model is independent, this step can by parallelized to speed up the generation of training data by splitting the simulation over many nodes on a computing cluster.

The trained HyDe-CNN architecture was able to achieve high accuracy for selecting among complex models of hybridization and admixture. Table 1 shows the overall prediction accuracy for both HyDe-CNN and the modified Flagel et al. network across input types and divergence scaling factors. As expected, prediction accuracy increases as the divergence scaling factor is increased, due to less conflicting signal in the data as a result of ILS. For both HyDe-CNN and the modified Flagel et al. architecture, the image encoding the minimum d_{XY} among pairs of species across the chromosome had the highest overall prediction accuracy across divergence scaling factors. The two-channel minimum+mean CNN also had accuracy comparable to the CNN trained on just the minimum d_{XY}. Interestingly, the CNNs trained on images of mean d_{XY} had much lower training and prediction accuracy regardless of architecture.

All trained models across both architectures primarily struggled to correctly identify the admixture model, especially at the 0.5 CU divergence scaling. The confusion matrices in Figure 3 break this down

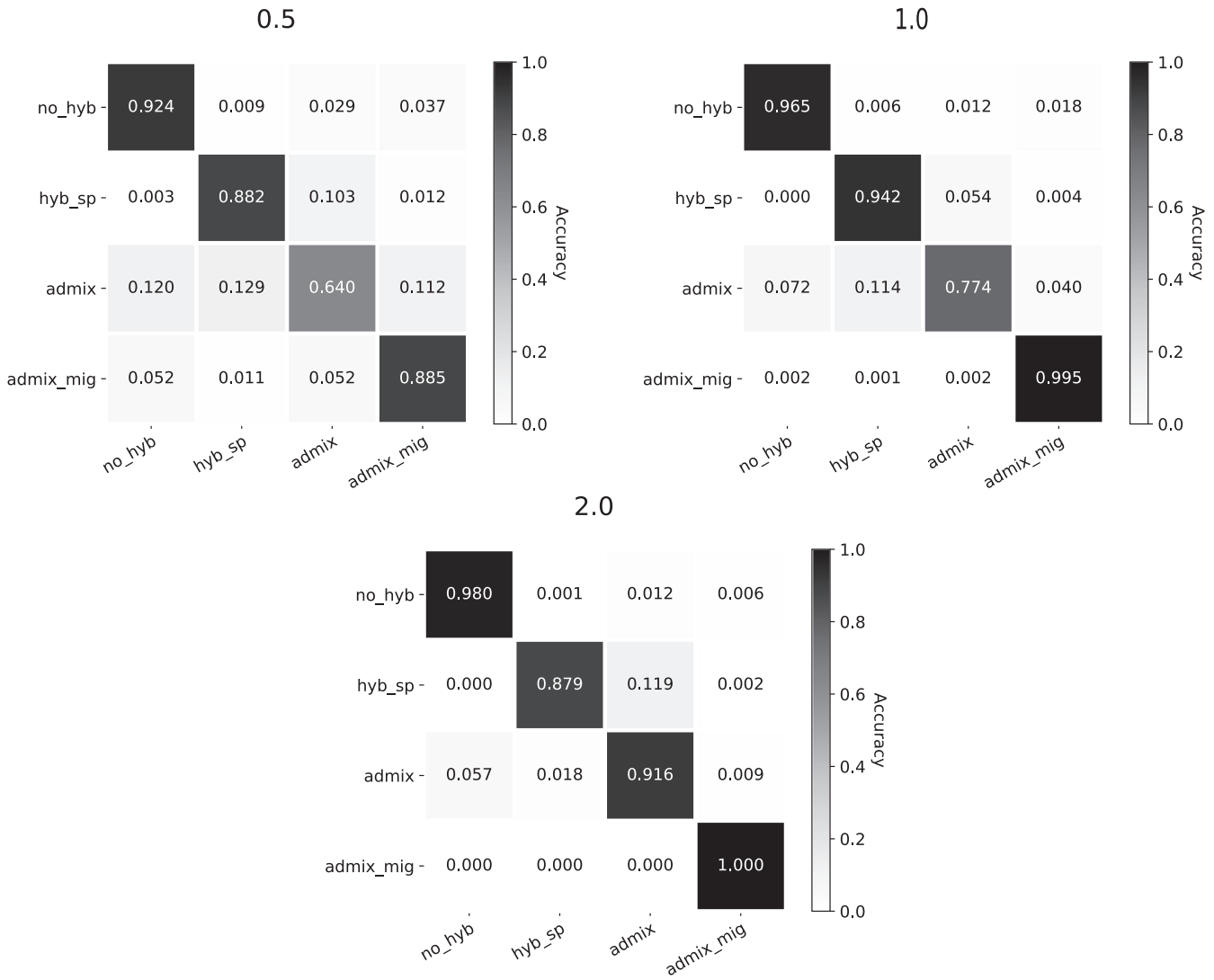

FIGURE 3 Confusion matrices for HyDe-CNN trained on minimum d_{XY}. Divergence scaling in coalescent units is noted above each plot. Model predictions were for 2500 simulated images from each model. In each plot, rows correspond to the true model, columns correspond to the predicted model, and entries in the matrix show the proportion of simulations coming from each true model that are predicted to come from any one of the four possible models

for the HyDe-CNN minimum d_{XY} network, showing the proportion of correctly predicted models (diagonals) as well as how different models were incorrectly predicted (off-diagonals) across the different divergence scalings. Precision and recall values for the HyDe-CNN minimum d_{XY} network are given in Table 2 and also show that false negatives (low recall) are primarily responsible for the low accuracy with the admixture model. The corresponding confusion matrix plots and precision/recall tables for the mean and minimum+mean HyDe-CNN predictions and the minimum and mean Flagel et al. network predictions are provided in the Supplemental Materials (Figures S1–S4 and Tables S2–S5).

3.2 | Properties of misidentified models

Using 10,000 additional simulations from each model to explore how different parameters affect the prediction accuracy of our CNN, we found that models were typically misidentified in areas of parameter space where it would be biologically difficult to distinguish between the true model and the chosen model. We also found that when the best model was not the generating model, the CNN tended to give it lower weight, demonstrating more uncertainty in the model selection process. The mean weight for incorrect models ranged from 0.639–0.741, compared with the mean weight given when the true model was selected, which ranged from 0.906–0.944. The pattern of model misidentification based on the input parameters was most clearly seen with the admixture model, which was the model with the highest error rate across the majority of our tests. In Figure 4, we plot the predicted model for all 10,000 data sets simulated under the admixture model, as well as the two most important parameters distinguishing this model from the others: the admixture fraction (*x*-axis) and the timing of admixture (*y*-axis). As the coalescent branch lengths increase (top

TABLE 2 Precision and recall of HyDe-CNN trained on images of minimum d_{XY} for model selection on independent test data. Divergence scaling is given in coalescent units (CU)

Divergence Scaling	Model	Precision	Recall
0.5	no_hyb	0.841	0.924
	hyb_sp	0.855	0.882
	Admix	0.777	0.640
	admix_mig	0.846	0.885
1.0	no_hyb	0.928	0.965
	hyb_sp	0.886	0.942
	Admix	0.919	0.774
	admix_mig	0.942	0.995
2.0	no_hyb	0.945	0.980
	hyb_sp	0.979	0.880
	admix	0.874	0.916
	admix_mig	0.983	1.000

to bottom rows), we see that the majority of images are correctly predicted to come from the admixture model with high prediction weight from the CNN. When an image is misclassified, it is typically predicted to come from either the no hybridization or hybrid speciation model. However, this misclassification is not random. The no-hybridization model is typically selected when the admixture fraction is close to 0. The hybrid speciation model is selected when the amount of admixture is higher (closer to 0.25) and when the timing of admixture is closer to the divergence between populations one and two. Similar plots for the no hybridization, hybrid speciation and admixture with gene flow models are in the Supplemental Materials (Figures S5–S7).

3.3 | Comparison with summary statistics

The random forest classifier trained using summary statistics showed the same pattern of increasing training accuracy as divergence scaling increased, having similar accuracy to the modified Flagel et al. network trained on minimum d_{XY} but lower accuracy when compared to HyDe-CNN across all input types (Table 1). However, it should be noted that our comparison here is between three simple statistics and two highly parameterized CNNs. The admixture model was again the most difficult to correctly classify, having a training misidentification rate ranging from 0.598 for the 0.5 CU simulations to 0.243 for the 2.0 CU simulations. Plotting the calculated test statistics across the models showed that there is considerable overlap in their values, which makes sense given that they are similar to one another, but also likely explains the poor prediction accuracy, especially at high levels of ILS (Figures S8–S10). Of the three statistics used, the variable importance metric reported by abcrf showed that the D_p statistic was the most informative for performing classification. Precision and recall values for the reserved test data across divergence scaling factors are given in Table S6.

3.4 | Heliconius butterflies

Independent test data run through the trained HyDe-CNN architecture for *Heliconius* showed that all input types (minimum, mean, minimum+mean) could distinguish among the three models with high accuracy (Figure S11). Using the trained networks to test for the mode of admixture between *mel*-W and *cyd* using chromosome five and 100 random samples of 5 individuals from each of the three ingroup populations (*mel*-E, *mel*-W, *cyd*) showed that the ancient admixture model was the predicted model for 98% of the minimum d_{XY} images, 100% of the mean d_{XY} images and 100% of the minimum + mean images. The average model weight across the 100 runs for the admixture model was 0.885 for the minimum d_{XY} network, 0.988 for the mean d_{XY} network and 0.991 for the minimum+mean d_{XY} network.

4 | DISCUSSION

Compared to more traditional, phylogenetic summary statistics, as well as a previously employed CNN architecture (Flagel et al. 2018), our HyDe-CNN approach produced model predictions with the highest levels of accuracy, precision and recall, especially at the shortest divergence scaling (0.5 CU) where the phylogenetic signal for hybridization is expected to be the most noisy. In particular, the networks that included minimum d_{XY} performed the best for predicting patterns of admixture. This makes sense given that hybridization leads to coalescent events that are more recent than the divergence times between population pairs, which is more likely to be captured by the minimum coalescent time between sampled chromosomes. Inferences of species divergence times and hybridization based on minimum coalescent times (Kubatko, 2009; Kubatko et al. 2009) or deviations from expected coalescent times (Joly et al. 2009) provided some of the first implementations of phylogenetic tests for hybridization. Using minimum d_{XY} within windows across the genome with HyDe-CNN therefore builds directly on these early ideas, providing a way to encode not only summaries of coalescent times but also their correlation across the genome due to linkage.

In our testing of HyDe-CNN, we deliberately chose to sample parameters from distributions rather than picking a small set of values to use for our simulations. In this way, we were able to test the robustness of the CNN's predictions across a wide swath of parameter space, showing that our method is capable of accurately selecting among competing hybridization models. For tests on empirical data, the ability to incorporate uncertainty for certain parameters is especially appealing, particularly for organisms where not as much is known about their evolutionary history. Our *Heliconius* analysis illustrates this flexibility well as we put distributions on the timing and magnitude of admixture but used fixed values for the rest of the input parameters. Nevertheless, the trained network showed high power for selecting among the candidate models (Figure S11) and overwhelmingly supported the ancient admixture model when analysing data from chromosome five, a result that agrees

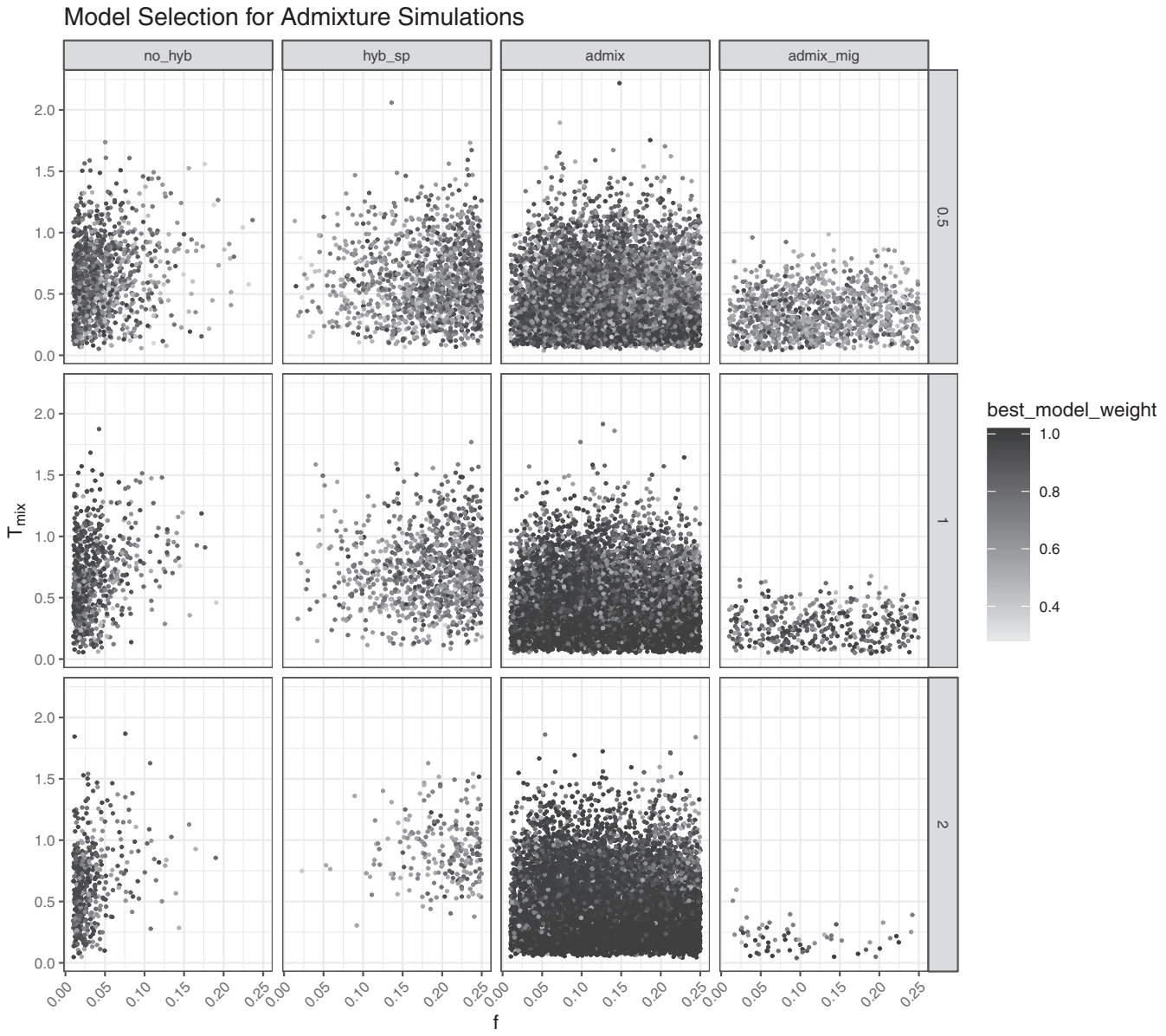

FIGURE 4 Scatterplot of model selection results for test simulations with the admixture model. Columns in the plot grid represent the predicted model, and rows represent the different divergence scaling factors in coalescent units. For each individual plot, the x-axis shows the fraction of admixture (*f*), the y-axis shows the timing of admixture (T_{mix}), and each dot represents the prediction for a single simulation (10,000 total). The shading of dots from light to dark corresponds to the predicted model weight (low to high, respectively)

with previous work (Martin et al. 2019). Current likelihood-free approaches such as ABC also allow for incorporating uncertainty when estimating evolutionary parameters and performing model selection (Beaumont et al. 2002). However, obtaining estimates of parameters with ABC relies on sampling all parameters in proportion to their approximate posterior probability, which can require a huge amount of simulation (Beaumont, 2010, the 'curse of dimensionality'). Methods that combine machine learning and ABC to reduce the amount of needed simulation are also being developed (Estoup et al. 2018; Pudlo et al. 2016) and are currently being used for applications in model selection for species delimitation (Smith & Carstens, 2020) and demographic inference (Mondal et al. 2019; Smith et al. 2017). Incorporating uncertainty estimates in convolutional neural networks are also beginning to emerge (Laumann et al., 2018) and are a potentially promising direction for extending our approach.

Another desirable feature of our HyDe-CNN method is that the trained network failed to correctly classify models in ways that reflect our biological expectations. While it may seem odd to highlight our method's failure, it is important to emphasize the interpretability of HyDe-CNN's predictions based on the parameters of the simulations that are used to train it. The best example of this is the method's difficulty with predicting the admixture model. The parameter bounds that we placed on this model for the timing and amount of admixture were chosen to approach the bounds of both the no hybridization and hybrid speciation models. With the no hybridization model, only admixture scenarios with low amounts of genetic

exchange were incorrectly inferred to come from a model with no admixture (Figure 4; first column). Similarly, when admixture models were mistakenly inferred for the hybrid speciation model, it was when the amount of admixture was close to 25%, the lower bound for γ in the hybrid speciation model (Figure 4; third column). Failing to correctly select the admixture model in these cases, particularly when ILS is high (0.5 CU simulations), as well as selecting the most biologically similar model rather than a random model, suggests to us that HyDe-CNN is able to successfully model real biology. Furthermore, the fact that HyDe-CNN typically gives less weight to incorrectly selected models suggests that the trained network is able to capture some aspects of uncertainty in the model selection process. Future work to decode the impact of the intermediate layers of the network, as well as determining which parts of the input images are most informative for model selection, will hopefully help to further uncover how biology plays into the model selection process for CNNs.

4.1 | Limitations and future directions

One important aspect of inferring hybridization that we largely ignore here is the presence of additional sources of heterogeneity across the genome caused by selection (direct or indirect) or by demographic changes such as bottlenecks. These phenomena will alter patterns of genetic diversity both within and between species in ways that could mislead inferences of hybridization when a CNN is trained on input data that are simulated without such forces. While our work focuses primarily on detecting hybridization at phylogenetic timescales, rather than on capturing specific genomic features resulting from the speciation process, events such as adaptive introgression or the evolution of genetic incompatibilities (i.e. Bateson–Dobzhansky–Muller incompatibilities; Dobzhansky, 1937; Muller, 1942) will leave distinctive and localized changes in between-species patterns of divergence. Indeed, this is why we chose to test HyDe-CNN on *Heliconius* chromosome five which has a fairly consistent signal of admixture across its length (excluding the chromosome ends) and does not contain any of the wing patterning loci that show signals for selection (Dasmahapatra et al. 2012; Martin et al. 2013, 2019). Future work using a more complex simulator such as SLiM 3 (Haller & Messer, 2019), which can incorporate selection, could provide a means to generate simulated training data appropriate for modelling more realistic patterns of genomic heterogeneity when inferring hybridization. Other techniques in deep learning, such as image segmentation for classifying portions of images (Long et al. 2015), could also eventually be employed to locate introgressed regions affected by adaptive introgression or to identify genetic incompatibilities.

Exploring more ways to extract information from genomic data using deep learning is likely to continue being an active area of research not only in terms of designing neural network architectures but also for thinking about how to represent genetic variation. The way that we represent genomic data as an image differs from other approaches that use the actual nucleotide alignment or genotype matrix as their input for CNN training (Battey et al., 2020; Flagel et al. 2018; Suvorov et al. 2019). We chose our data representation because we believed it to be an intuitive summary of pairwise coalescence times between species organized by the pattern of divergence in the underlying phylogeny. Extensions of this representation would be simple and could include different summaries of d_{XY}, such as quantiles of the pairwise distribution, as well as more summary statistics, which would be included as additional channels in the input image. There is, however, some arbitrariness in how we represent genomic data as an image. This includes both the choice of how many windows to use for dividing up a chromosome, as well as the ordering of pairwise comparisons for species pairs with equal phylogenetic distances (e.g. $P_1{\times}P_3$ and $P_2{\times}P_3$). For the choice of window number, a possible solution would be to train multiple networks with different window sizes to see whether it affects downstream inferences. It might also be possible to train a single network with different numbers of windows by padding the input images, similar to how variable numbers of SNPs are handled in CNNs trained on binary genotype matrices (Flagel et al. 2018). Dealing with the arbitrary ordering of species with equal phylogenetic distance could be handled by generating all possible orderings and using them during training. The use of exchangeable networks (Chan et al. 2018), a concept developed to deal with the arbitrary order of individuals within populations, could also be a potential avenue to explore for dealing with redundant phylogenetic information encoded by species pairs.

As we mentioned briefly above, our network architecture is both smaller and simpler than other CNNs used in population genetics. On the one hand, this makes the network easy to train, does not require the use of a GPU and is less likely to lead to overfitting, all while still generalizing to making predictions on independent test data. On the other hand, it is possible that a larger network could provide more power to tease apart subtle differences between similar models (e.g. admixture versus hybrid speciation) and could also allow us to expand the modest set of models that we tested. More sophisticated and larger network architectures, such as exchangeable (Chan et al. 2018) and recurrent (Adrion et al. 2020) neural networks, have led to powerful inferences of recombination landscapes and could potentially help to better leverage the linkage information encoded by our representation. However, the focus of these larger networks is usually centred around the estimation of continuous parameters rather than on model selection. Therefore, similar network architectures could also potentially be used to extend our work to directly estimate the timing and amount of hybridization instead of distinguishing among a predetermined set of hybridization scenarios. As more researchers begin working with deep learning models, we look forward to the continued testing and refinement of methods for not only estimating patterns of hybridization but for making other phylogenomic inferences as well.

ACKNOWLEDGEMENTS

The authors thank M. L. Smith for the invitation to submit this study for the special issue on Machine Learning in Molecular

Ecology. We also thank J. E. James and S. Martin for help with processing and modelling the *Heliconius* data, as well as the *Heliconius* community in general for making their excellent genomic resources publicly available. In addition, we are grateful to members of the Barker and Gutenkunst laboratories, F. Austerlitz, C. Roux, D. Lawson and an anonymous reviewer for comments that helped to improve this manuscript. This work was supported by a National Science Foundation Postdoctoral Research Fellowship in Biology (IOS-1811784 to P.D.B.) and by the National Institute of General Medical Sciences of the National Institutes of Health (R01GM127348 to R.N.G.).

AUTHOR CONTRIBUTIONS

P.D.B. designed the study with guidance from M.S.B. and R.N.G. P.D.B. performed the simulations and data analyses. P.D.B. wrote the initial manuscript draft with input from M.S.B. and R.N.G. All authors read and approved the final manuscript.

DATA AVAILABILITY STATEMENT

Code for all simulations and analyses is available on GitHub (https://github.com/pblischak/hyde-cnn.git) with additional documentation on ReadTheDocs (https://pblischak.github.io/hyde-cnn/). All simulated input images and trained models are deposited on Dryad [https://doi.org/10.5061/dryad.63xsj3v0r; Blischak et al., 2020).

ORCID

Paul D. Blischak https://orcid.org/0000-0001-9177-8958
Michael S. Barker https://orcid.org/0000-0001-7173-1319

REFERENCES

Abadi, M., Agarwal, A., & Barham, P. (2016). TensorFlow: Large-scale machine learning on heterogeneous distributed systems. *arXiv preprint arXiv:1603.04467*.

Adrion, J. R., Galloway, J. G., & Kern, A. D. (2020). Predicting the landscape of recombination using deep learning. *Molecular Biology and Evolution*, *37*, 1790–1808.

Agarap, A. F. (2018). Deep learning using rectified linear units (ReLU). *arXiv preprint arXiv:1803.08375*.

Amarasinghe, S. L., Su, S., Dong, X., Zappia, L., Ritchie, M. E., & Gouil, Q. (2020). Opportunities and challenges in long-read sequencing data analysis. *Genome Biology*, *21*, 30.

Anderson, E. (1949). *Introgressive hybridization*. John Wiley.

Anderson, E., & Stebbins, G. L. (1954). Hybridization as an evolutionary stimulus. *Evolution*, *8*, 378–388.

Bates, H. W. (1862). Contributions to an insect fauna of the amazon valley. Lepidoptera: Heliconidae. *Transactions of the Linnean Society of London*, *23*, 495–566.

Battey, C. J., Ralph, P. L., & Kern, A. D. (2020). Predicting geographic location from genetic1variation with deep neural networks. *bioRxiv*, https://doi.org/10.1101/2019.12.11.872051

Beaumont, M. A. (2010). Approximate Bayesian computation in evolution and ecology. *Annual Review of Ecology, Evolution, and Systematics*, *41*, 379–406.

Beaumont, M. A., Zhang, W., & Balding, D. J. (2002). Approximate Bayesian computation in population genetics. *Genetics*, *162*, 2025–2035.

Blischak, P. D., Barker, M. S., & Gutenkunst, R. N. (2020). Data for: Chromosome-scale inference of hybrid speciation and admixture with convolutional neural networks. *Dryad*, https://doi.org/10.5061/dryad.63xsj3v0r

Blischak, P. D., Chifman, J., Wolfe, A. D., & Kubatko, L. S. (2018). HyDe: A Python package for genome-scale hybridization detection. *Systematic Biology*, *67*, 821–829.

Chan, J., Perrone, V., Spence, J., Jenkins, P., Mathieson, S., & Song, Y. (2018). A likelihood-free inference framework for population genetic data using exchangeable neural networks. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, & R. Garnett (Eds.), *Advances in neural information processing systems 31* (pp. 8594–8605). Curran Associates Inc.

Chen, L., Wolf, A. B., Fu, W., Li, L., & Akey, J. M. (2020). Identifying and interpreting apparent Neanderthal ancestry in African individuals. *Cell*, *180*, 677–687.

Dasmahapatra, K. K., Walters, J. R., & Briscoe, A. D. (2012). Butterfly genome reveals promiscuous exchange of mimicry adaptations among species. *Nature*, *487*, 94–98.

Davey, J. W., Barker, S. L., & Rastas, P. M. (2017). No evidence for maintenance of a sympatric *Heliconius* species barrier by chromosomal inversions. *Evolution Letters*, *1*, 138–154.

Dobzhansky, T. (1937). *Genetics and the origin of species*. Columbia University Press.

Durand, E. Y., Patterson, N., Reich, D., & Slatkin, M. (2011). Testing for ancient admixture between closely related populations. *Molecular Biology and Evolution*, *28*, 2239–2252.

Dutheil, J. Y., Ganapathy, G., Hobolth, A., Mailund, T., Uyenoyama, M. K., & Schierup, M. H. (2009). Ancestral population genomics: The coalescent hidden Markov model approach. *Genetics*, *183*, 259–274.

Eaton, D. A. R., & Ree, R. H. (2013). Inferring phylogeny and introgression using RADseq data: an example from the flowering plants (*Pedicularis*: Orobanchaceae). *Systematic Biology*, *62*, 689–706.

Edelman, N. B., Frandsen, P. B., Miyagi, M., Clavijo, B., Davey, J., Dikow, R. B., García-Accinelli, G., Van Belleghem, S. M., Patterson, N., Neafsey, D. E., Challis, R., Kumar, S., Moreira, G. R. P., Salazar, C., Chouteau, M., Counterman, B. A., Papa, R., Blaxter, M., Reed, R. D., … Mallet, J. (2019). Genomic architecture and introgression shape a butterfly radiation. *Science*, *366*, 594–599.

Elworth, R. A. L., Allen, C., Benedict, T., Dulworth, P., & Nakhleh, L. (2018). DGEN: A test statistic for detection of general introgression scenarios. *bioRxiv*, https://doi.org/10.1101/348649

Estoup, A., Raynal, L., Verdu, P., & Marin, J. M. (2018). Model choice using approximate Bayesian computation and random forests: analyses based on model grouping to make inferences about the genetic history of Pygmy human populations. *Journal de la Société Française de Statistique*, *159*, 167–190.

Flagel, L., Brandvain, Y., & Schrider, D. R. (2018). The unreasonable effectiveness of convolutional neural networks in population genetic inference. *Molecular Biology and Evolution*, *36*, 220–238.

Forsythe, E. S., Sloan, D. B., & Beilstein, M. A. (2020). Divergence-based introgression polarization. *Genome Biology and Evolution*, *12*, 463–478.

Green, R. E., Krause, J., Briggs, A. W., Maricic, T., Stenzel, U., Kircher, M., Patterson, N., Li, H., Zhai, W., Fritz, M. H. Y., Hansen, N. F., Durand, E. Y., Malaspinas, A. S., Jensen, J. D., Marques-Bonet, T., Alkan, C., Prufer, K., Meyer, M., Burbano, H. A., … Paabo, S. (2010). A draft sequence of the Neandertal genome. *Science*, *328*, 710–722.

Griffiths, R. C., & Marjoram, P. (1997). An ancestral recombination graph. In P. Donnelly & S. Tavare (Eds.), *Progress in Population Genetics and Human Evolution* (pp. 257–270). Springer.

Hahn, M. W., & Hibbins, M. S. (2019). A three-sample test for introgression. *Molecular Biology and Evolution*, *36*, 2878–2882.

Haller, B. C., & Messer, P. W. (2019). SLiM 3: Forward genetic simulations beyond the Wright-Fisher model. *Molecular Biology and Evolution*, *36*, 632–637.

Hamlin, J. A. P., Hibbins, M. S., & Moyle, L. C. (2020). Assessing biological factors affecting postspeciation introgression. *Evolution Letters*, *4*, 137–154.

Harris, K., & Nielsen, R. (2016). The genetic cost of Neanderthal introgression. *Genetics*, *203*, 881–891.

Hibbins, M. S., & Hahn, M. W. (2019). The timing and direction of introgression under the multispecies network coalescent. *Genetics*, *211*, 1059–1073.

Hobolth, A., Christensen, O., Mailund, T., & Schierup, M. (2007). Genomic relationships and speciation times of human, chimpanzee, and gorilla inferred from a coalescent hidden Markov model. *PLoS Genetics*, *3*, e7.

Hudson, R. R. (1983). Properties of a neutral allele model with intragenic recombination. *Theoretical Population Biology*, *23*, 183–201.

Joly, S., McLenachan, P. A., & Lockhart, P. J. (2009). A statistical approach for distinguishing hybridization and incomplete lineage sorting. *The American Naturalist*, *174*, e54–e70.

Juric, I., Aeschbacher, S., & Graham, C. (2016). The strength of selection against Neanderthal introgression. *PLoS Genetics*, *12*, e1006340.

Kelleher, J., Etheridge, A. M., & McVean, G. (2016). Efficient coalescent simulation and genealogical analysis for large sample sizes. *PLoS Computational Biology*, *12*, e1004842.

Kelleher, J., Thornton, K. R., Ahsander, J., & Ralph, P. L. (2018). Efficient pedigree recording for fast population genetics simulation. *PLoS Computational Biology*, *14*, e1006581.

Kelleher, J., Wong, Y., Wohns, A. W., Fadil, C., Albers, P. K., & McVean, G. (2019). Inferring whole-genome histories in large population datasets. *Nature Genetics*, *51*, 1330–1338.

Kingma, D. P., & Ba, J. (2017). *Adam: A method for stochastic optimization*.

Kingman, J. F. C. (1982). On the genealogy of large populations. *Journal of Applied Probability*, *19*, 27–43.

Kubatko, L. S. (2009). Identifying hybridization events in the presence of coalescence via model selection. *Systematic Biology*, *58*, 478–488.

Kubatko, L. S., Carstens, B. C., & Knowles, L. L. (2009). STEM: species tree estimation using maximum likelihood for gene trees under coalescence. *Bioinformatics*, *25*, 971–973.

Kubatko, L. S., & Chifman, J. (2019). An invariants-based method for efficient identification of hybrid species from large-scale genomic data. *BMC Evolutionary Biology*, *19*, 112.

Kuhn, M. (2008). Building predictive models in R using the caret package. *Journal of Statistical Software*, *28*, 1–26.

Laumann, F., Shridhar, K., & Maurin, A. L. (2018). Bayesian convolutional neural networks. *arXiv preprint arXiv:1806.05978*.

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, *521*, 436–444.

LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, *86*, 2278–2324.

Li, H., & Durbin, R. (2011). Inference of human population history from individual whole-genome sequences. *Nature*, *475*, 493–496.

Li, H., Handsaker, B., Wysoker, A., Fennell, T., Ruan, J., Homer, N., Marth, G., Abecasis, G., & Durbin, R. (2009). The sequence alignment/map format and SAMtools. *Bioinformatics*, *25*, 2078–2079.

Liu, K. J., Dai, J., Truong, K., Song, Y., Kohn, M. H., & Nakhleh, L. (2014). An HMM-based comparative genomic framework for detecting introgression in eukaryotes. *PLoS Computational Biology*, *10*, e1003649.

Long, J., Shelhamer, E., & Darrell, T. (2015). Fully convolutional networks for semantic segmentation. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015*, 3431–3440.

Mailund, T., Halager, A. E., Westergaard, M., Dutheil, J. Y., Munch, K., Andersen, L. N., Lunter, G., Prüfer, K., Scally, A., Hobolth, A., & Schierup, M. H. (2012). A new isolation with migration model along complete genomes infers very different divergence processes among closely related great ape species. *PLoS Genetics*, *8*, e1003125.

Martin, S. H., Dasmahapatra, K. K., & Nadeau, N. J. (2013). Genome-wide evidence for speciation with gene flow in *Heliconius* butterflies. *Genome Research*, *23*, 1817–1828.

Martin, S. H., Davey, J. W., Salazar, C., & Jiggins, C. D. (2019). Recombination rate variation shapes barriers to introgression across butterfly genomes. *PLoS Biology*, *17*, e2006288.

McVean, G. A. T., & Cardin, N. J. (2005). Approximating the coalescent with recombination. *Philosophical Transactions of the Royal Society B*, *360*, 1387–1393.

Moest, M., Van Belleghem, S. M., James, J. E., Salazar, C., Martin, S. H., Barker, S. L., Moreira, G. R. P., Mérot, C., Joron, M., Nadeau, N. J., Steiner, F. M., & Jiggins, C. D. (2020). Selective sweeps on novel and introgressed variation shape mimicry loci in a butterfly adaptive radiation. *PLoS Biology*, *18*, e3000597.

Mondal, M., Bertranpetit, J., & Lao, O. (2019). Approximate Bayesian computation with deep learning supports a third archaic introgression in Asia and Oceania. *Nature Communications*, *10*, 246.

Muller, H. J. (1942). Isolating mechanisms, evolution, and temperature. *Biology Symposium*, *6*, 71–125.

Nadeau, N. J., Martin, S. H., Kozak, K. M., Salazar, C., Dasmahapatra, K. K., Davey, J. W., Baxter, S. W., Blaxter, M. L., Mallet, J., & Jiggins, C. D. (2013). Genome-wide patterns of divergence and gene flow across a butterfly radiation. *Molecular Ecology*, *22*, 814–826.

Nei, M. (1987). *Molecular evolutionary genetics*. Columbia University Press.

Nei, M., & Li, W. H. (1979). Mathematical model for studying genetic variation in terms of restriction endonucleases. *Proceedings of the National Academy of Sciences of the United States of America*, *76*, 5269–5273.

Pease, J. B., & Hahn, M. W. (2015). Detection and polarization of introgression in a five-taxon phylogeny. *Systematic Biology*, *64*, 651–662.

Pedregosa, F., Varoquaux, G., & Gramfort, A. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, *12*, 2825–2830.

Poplin, R., Chang, P.-C., Alexander, D., Schwartz, S., Colthurst, T., Ku, A., Newburger, D., Dijamco, J., Nguyen, N., Afshar, P. T., Gross, S. S., Dorfman, L., McLean, C. Y., & DePristo, M. A. (2018). A universal SNP and small-indel variant caller using deep neural networks. *Nature Biotechnology*, *36*, 983–987.

Pudlo, P., Marin, J. M., Estoup, A., Cornuet, J. M., Gautier, M., & Robert, C. P. (2016). Reliable ABC model choice via random forests. *Bioinformatics*, *32*, 859–866.

R Core Team. (2019). *R: A language and environment for statistical computing*. R Foundation for Statistical Computing.

Ralph, P., Thornton, K., & Kelleher, J. (2020). Efficiently summarizing relationships in large samples: A general duality between statistics of genealogies and genomes. *Genetics*, *215*(3), 779–797. https://doi.org/10.1534/genetics.120.303253

Schiffels, S., & Durbin, R. (2014). Inferring human population size and separation history from multiple genome sequences. *Nature Genetics*, *46*, 919–925.

Schrider, D. R., & Kern, A. D. (2018). Supervised machine learning for population genetics: a new paradigm. *Trends in Genetics*, *34*, 301–312.

Sheehan, S., & Song, Y. S. (2016). Deep learning for population genetic inference. *PLoS Genetics*, *12*, e1004845.

Smith, M. L., & Carstens, B. C. (2020). Process-based species delimitation leads to identification of more biologically relevant species. *Evolution*, *74*, 216–229.

Smith, M., Ruffley, M., Espíndola, A., Tank, D., Sullivan, J., & Carstens, B. (2017). Demographic model selection using random forests and the site frequency spectrum. *Molecular Ecology*, *26*, 4562–4573.

Speidel, L., Forest, M., Shi, S., & Myers, S. R. (2019). A method for genome-wide genealogy estimation for thousands of samples. *Nature Genetics*, *51*, 1321–1329.

Suvorov, A., Hochuli, J., & Schrider, D. R. (2019). Accurate inference of tree topologies from multiple sequence alignments using deep learning. *Systematic Biology*, *69*, 221–233.

Tajima, F. (1983). Evolutionary relationship of DNA sequences in finite populations. *Genetics*, *105*, 437–460.
Terhorst, J., Kamm, J., & Song, Y. (2017). Robust and scalable inference of population history from hundreds of unphased whole genomes. *Nature Genetics*, *49*, 303–309.
Van Der Walt, S., Colbert, S. C., & Varoquaux, G. (2011). The numpy array: a structure for efficient numerical computation. *Computing in Science & Engineering*, *13*, 22.
Watterson, G. A. (1975). On the number of segregating sites in genetical models without recombination. *Theoretical Population Biology*, *7*, 256–276.
Wickham, H. (2009). *ggplot2: elegant graphics for data analysis*. Springer.
Wright, S. (1931). Evolution in Mendelian populations. *Genetics*, *16*, 97–159.
Wright, S. (1943). Isolation by distance. *Genetics*, *28*, 114–138.

SUPPORTING INFORMATION

Additional supporting information may be found online in the Supporting Information section.

How to cite this article: Blischak PD, Barker MS, Gutenkunst RN. Chromosome-scale inference of hybrid speciation and admixture with convolutional neural networks. *Mol Ecol Resour.* 2021;21:2676–2688. https://doi.org/10.1111/1755-0998.13355