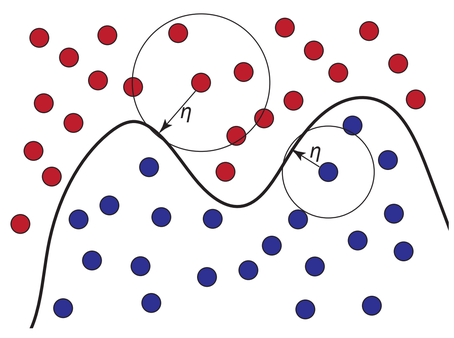
i trained i : Exploring adversarial training for population genomics

Jeffrey R. Adrion[1] and Andrew D. Kern[1]

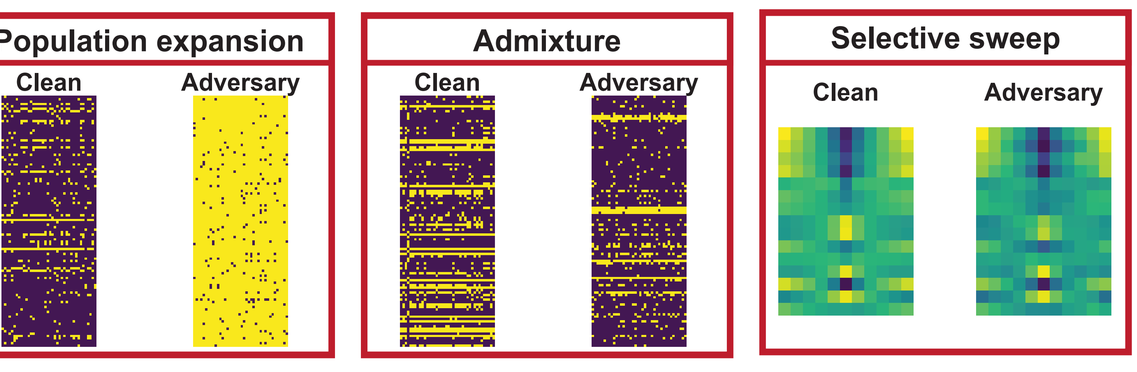
[1]Institute of Ecology and Evolution, University of Oregon, Eugene, OR

Introduction

There has been a recent explosion in the application of supervised machine learning methods within the fields of population genetics, genomics, and phylogenetics. These tools come with a unique set of constraints and potential hazards. Perhaps the most obvious of these limitations are the problems of overfitting and out-of-sample prediction, where the training set is a poor match to the test data. Here we explore training with the inclusion of adversarial examples—inputs crafted by making the smallest perturbation that results in a high-confidence misclassification of the example (right)—as a method to assess and potentially increase robustness to common model misspecifications.



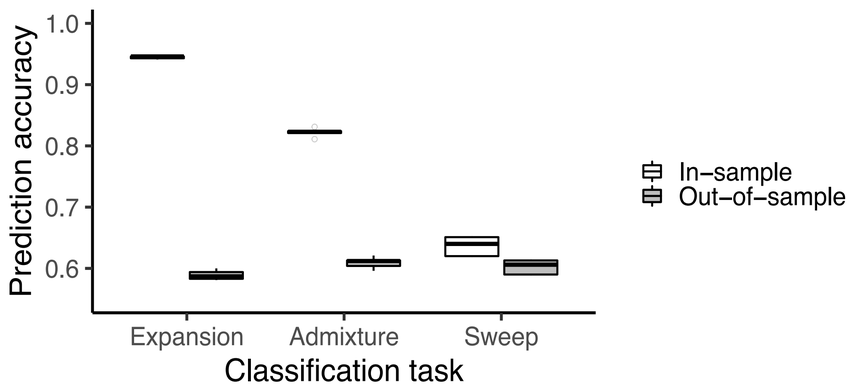
Classification tasks



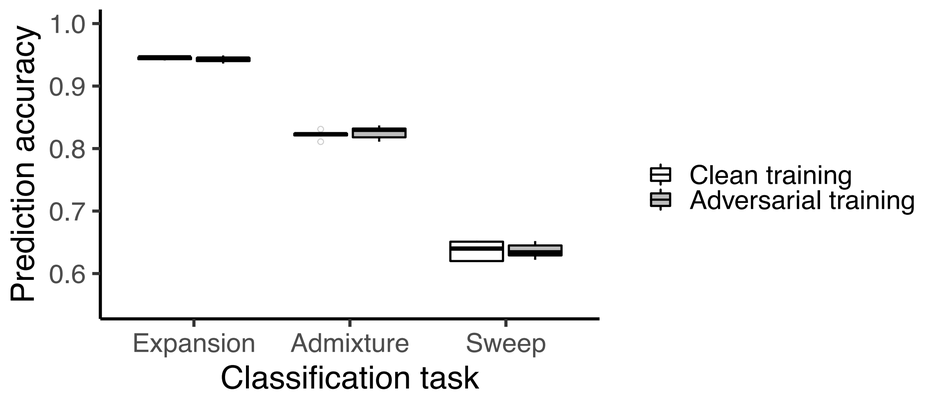
Adversarial training

We trained neural networks (GRUs and CNNs) for each of three classification tasks. We used the raw genotype matrix as input for classifying population expansions and admixture, and for selective sweeps we used a matrix of population genetic summary statistics. We then generated adversarial examples for each input (right). A second network was later trained on a mixture of clean and adversarial examples. Prediction accuracy was compared for examples matching the training set (in-sample) and for misspecified examples (out-out-sample).

Accuracy is significantly lower for out-of-sample prediction

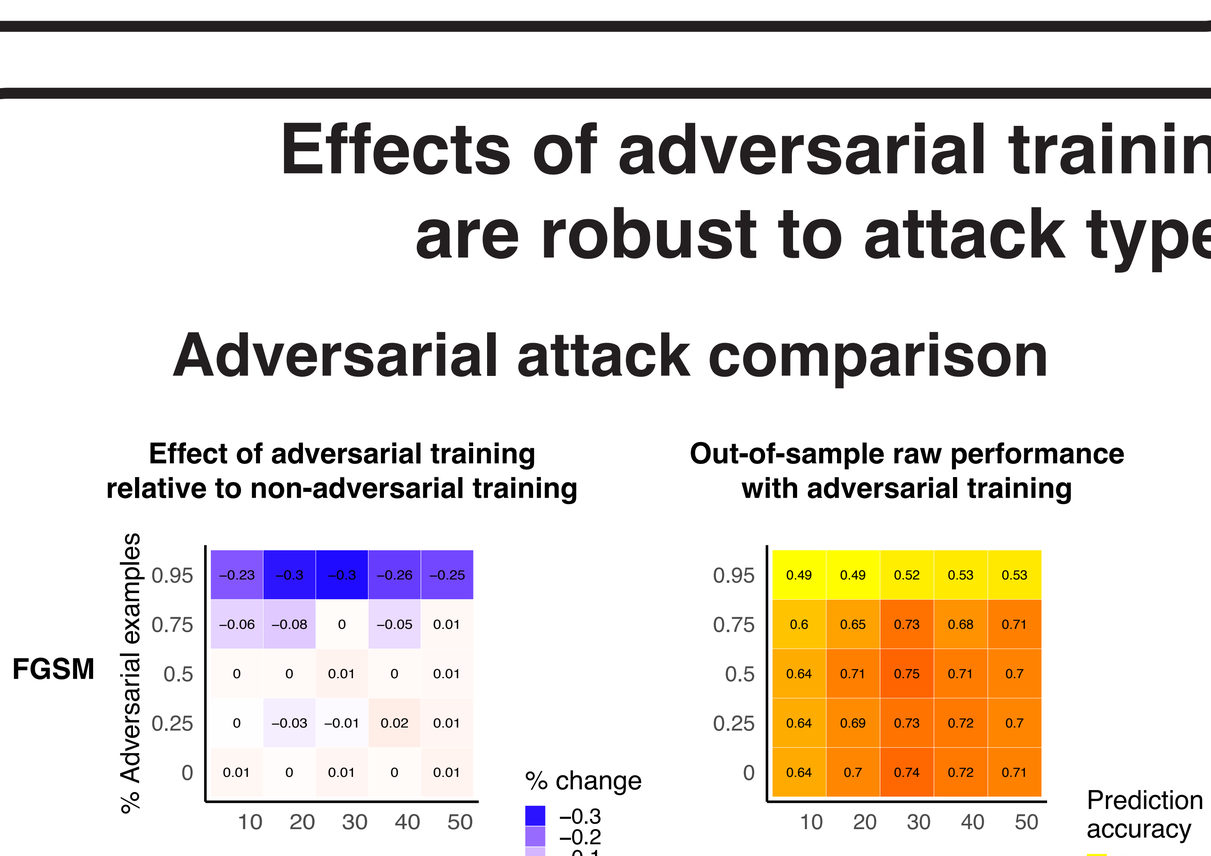


No effect of adversarial training for in-sample prediction

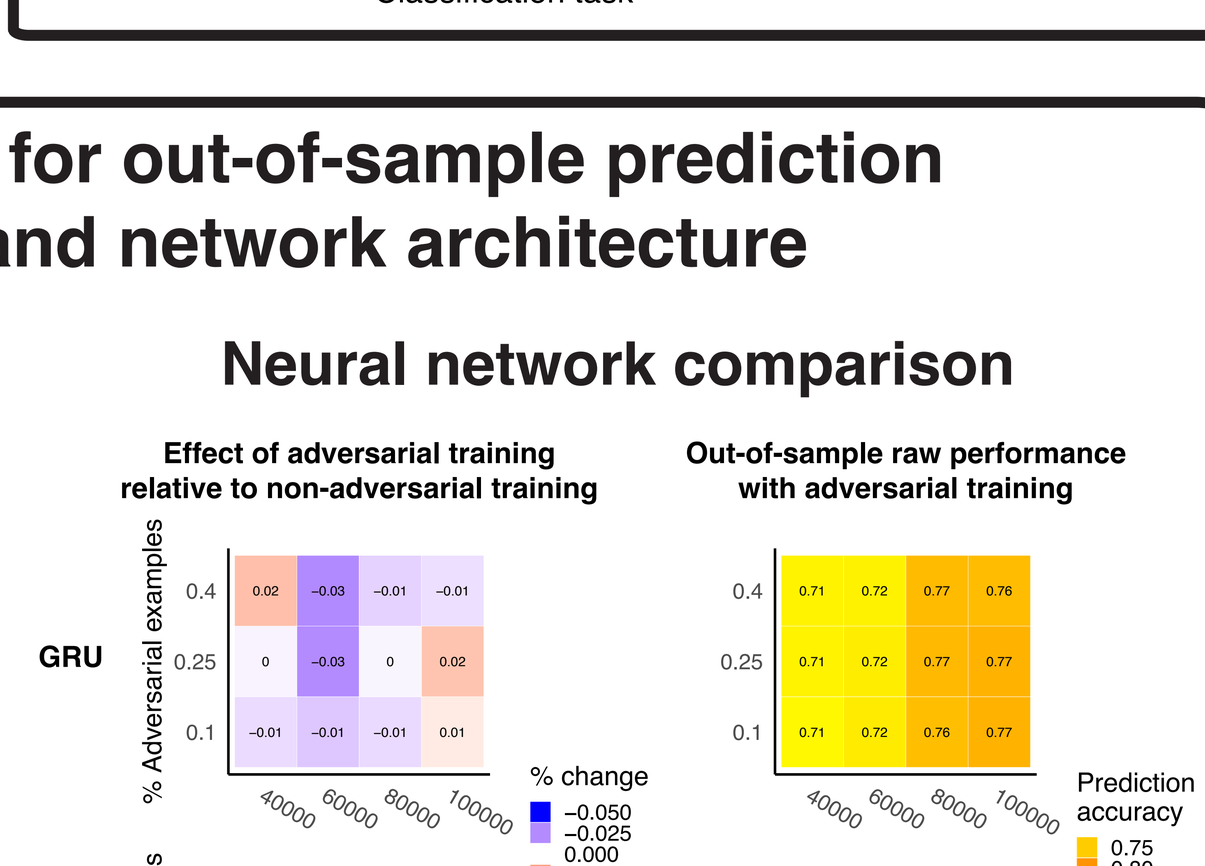


Effects of adversarial training for out-of-sample prediction are robust to attack type and network architecture

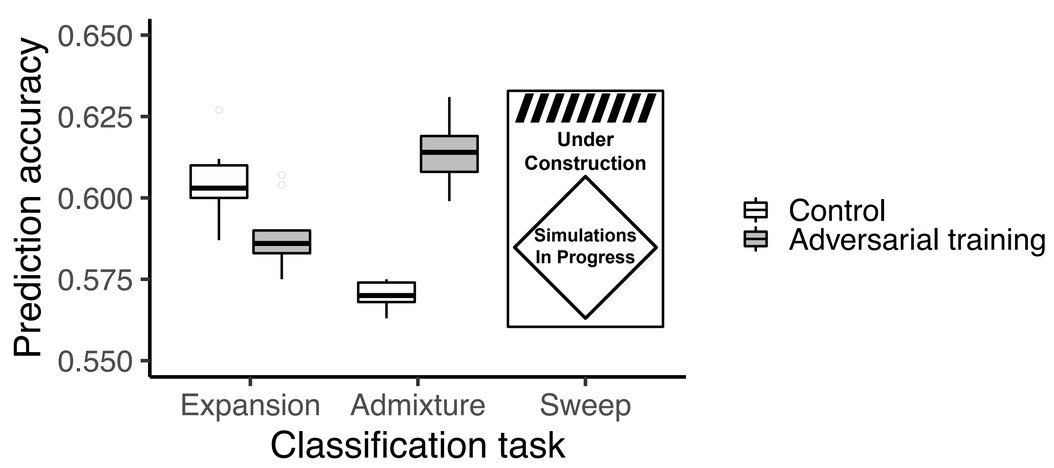
Adversarial attack comparison



Neural network comparison



Effects of adversarial training for out-of-sample prediction differ between tasks



Acknowledgments

We would like to thank members of the Kern and Ralph labs for their thoughtful comments. Simulations were generated with msprime (https://github.com/tskit-dev/msprime) and discoal (https://github.com/kern-lab/discoal). All adversarial attacks were completed using cleverhans (https://github.com/tensorflow/cleverhans). Lastly, we would like to thank Bad Brains for the soundtrack to making this poster and for inspiring its title.