

Machine learning in population and phylogenetics

Megan L Smith
Woods Hole Workshop on Molecular Evolution

Table of contents

1. Introduction and Motivation
2. Overview of Supervised Machine Learning Algorithms
3. Challenges and Future Directions
4. How and when should I use Machine Learning?
5. Useful Tools
6. Jupyter Notebook Example

Outline

Overview of Supervised Machine Learning Algorithms

Decision Trees

Fully Connected Neural Networks (FCNNs)

Convolutional Neural Networks (CNNs)

Graphical Neural Networks

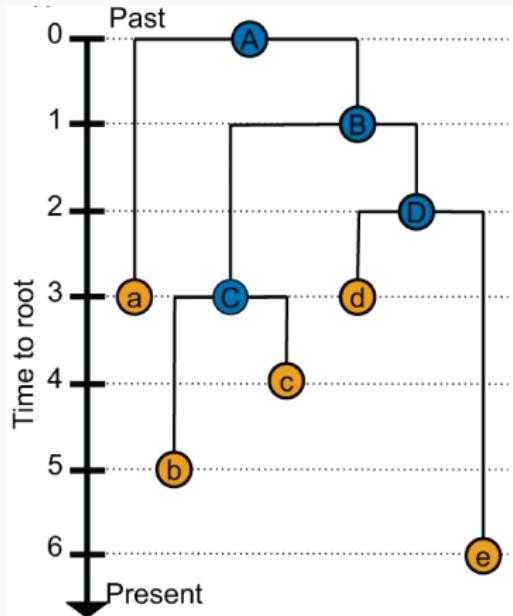
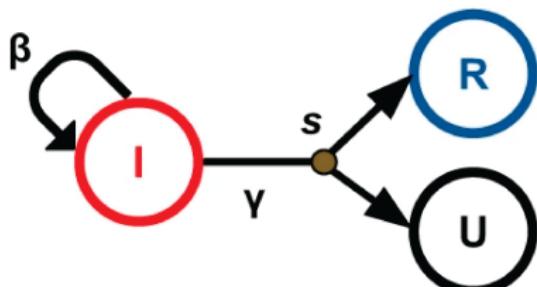
Recurrent Neural Networks

Generative Models

Overview of Algorithms

A motivating example for GNNs

$$R_0 = \beta/\gamma$$

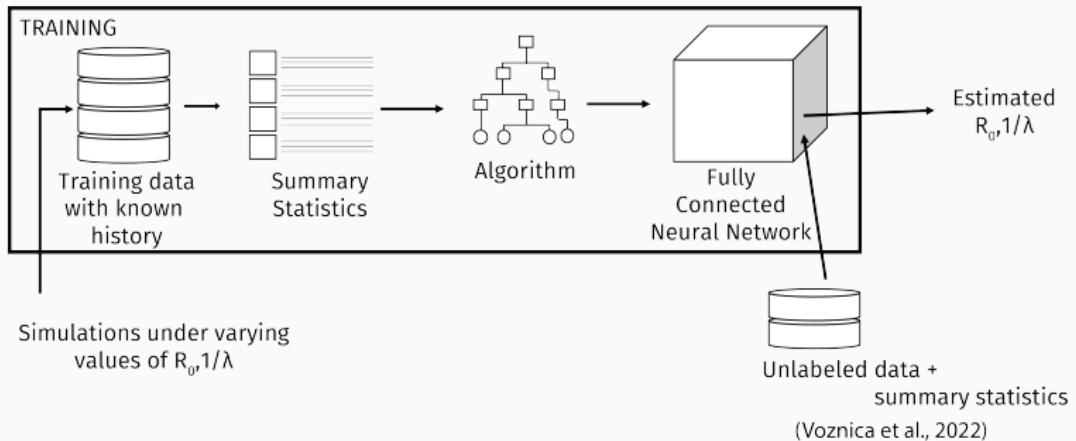


Figures 1, 2 from Voznica et al., 2022

A motivating example for GNNs

- How could you use machine learning to estimate these parameters?
- Available data:
 1. sequence data
 2. phylogeny

A motivating example for GNNs



Summary statistics: 26 measures of branch lengths, 8 measures of tree topology, 9 measures of the number of lineages through time, and 40 coordinates representing the lineage-through-time (LTT) plot.

A motivating example for GNNs

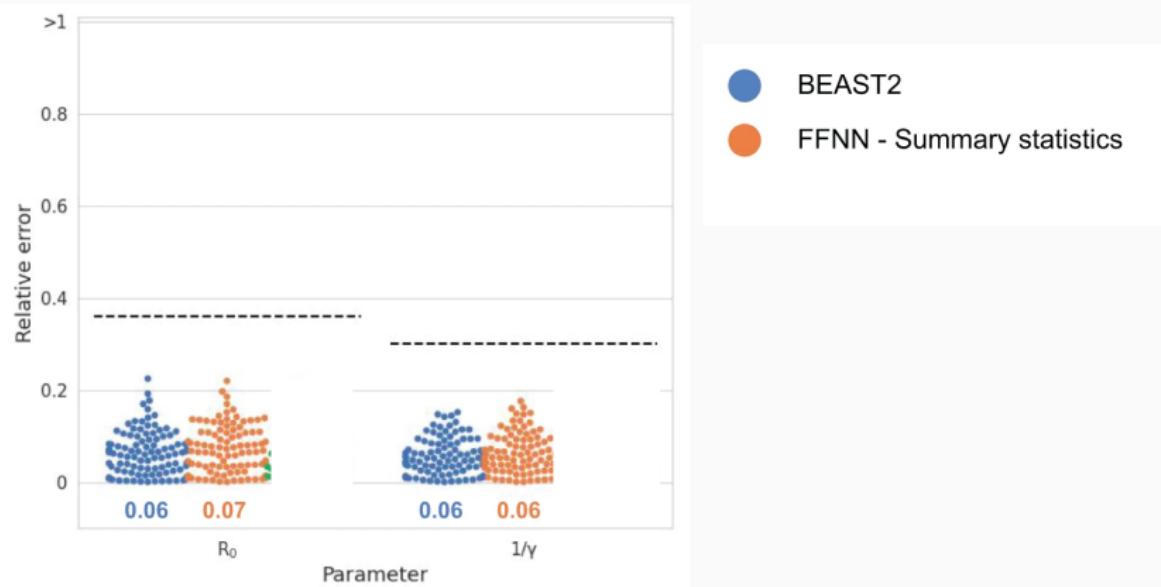


Figure 3 from Voznica et al., 2022

A motivating example for GNNs

Compact Bijective Ladderized Vector encoding

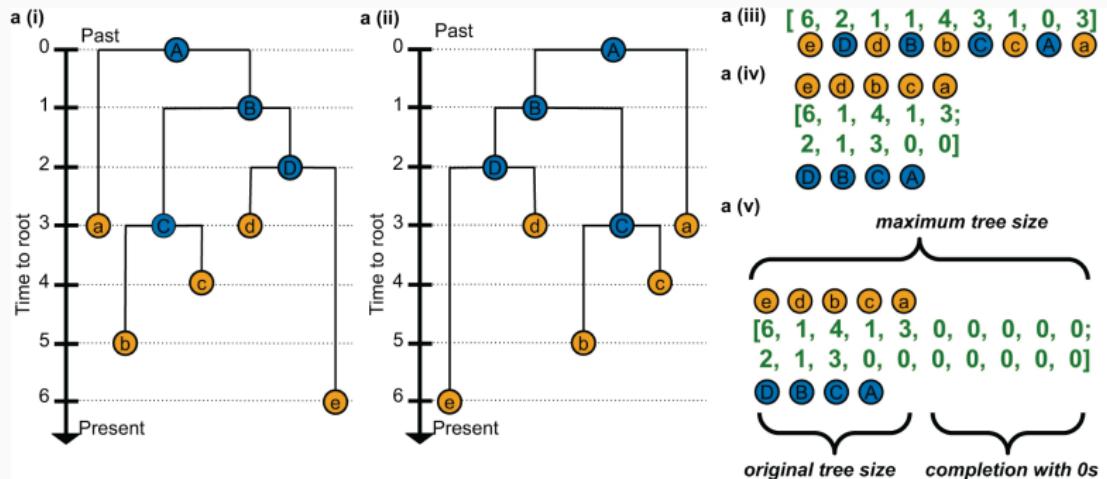
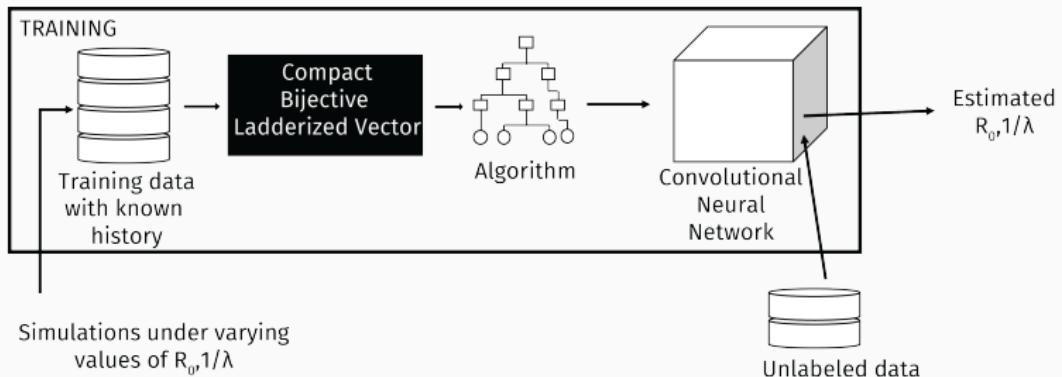


Figure 2 from Voznica et al., 2022

A motivating example for GNNs



(Voznica et al., 2022)

A motivating example for GNNs

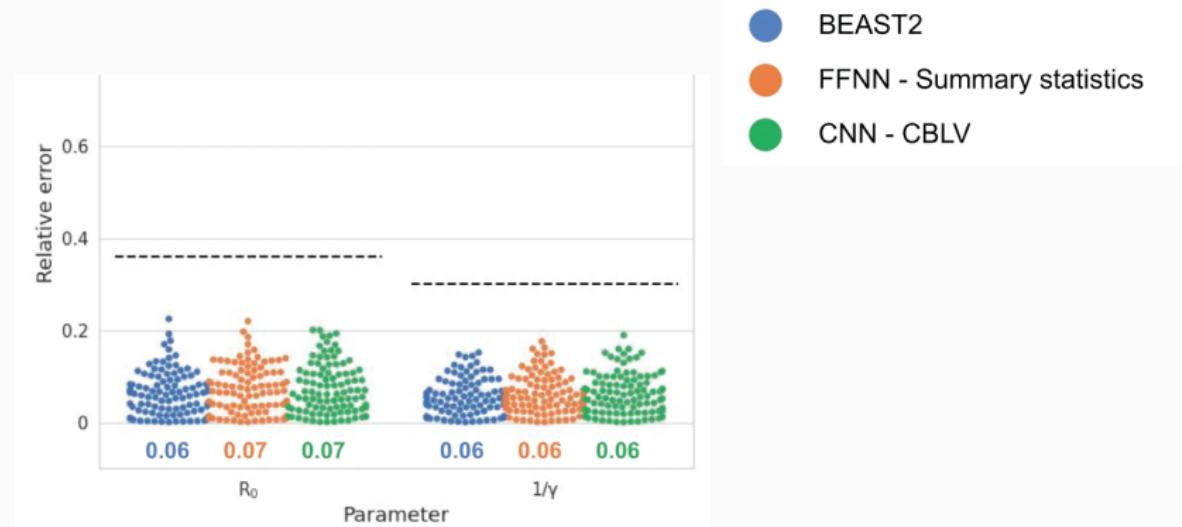


Figure 3 from Voznica et al., 2022

A motivating example for GNNs

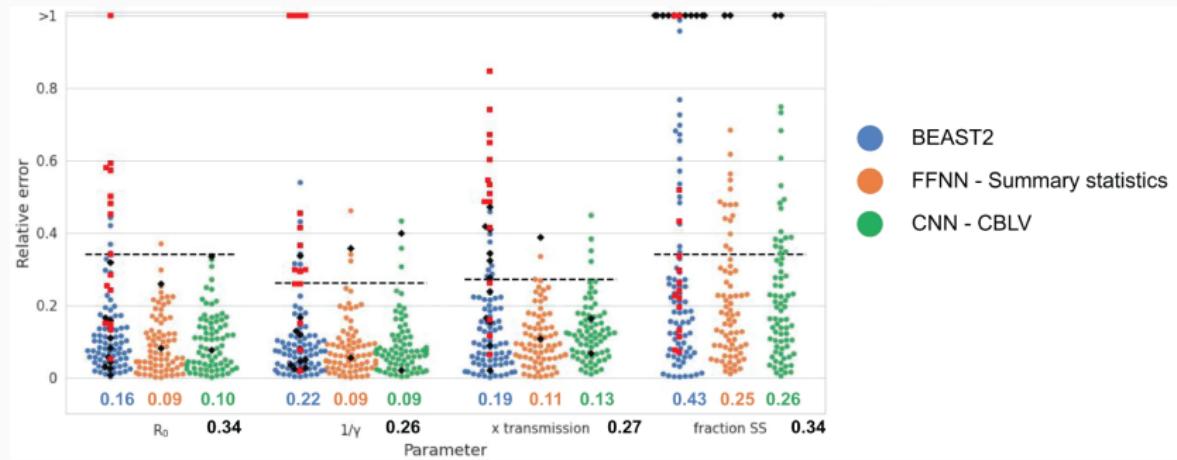


Figure 3 from Voznica et al., 2022

A motivating example for GNNs

A similar goal: estimating speciation λ and extinction μ rates from a phylogeny (Lajaaiti, et al., 2023).

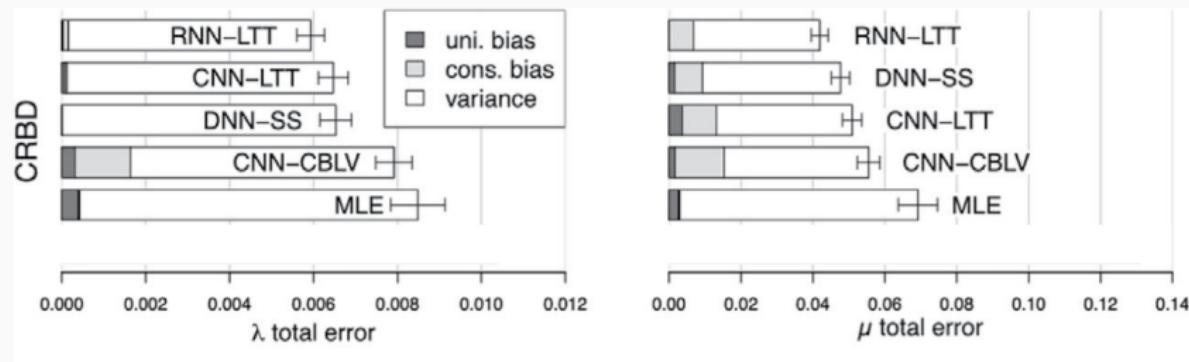


Figure 3 from Latjaaiti et al., 2023

A motivating example for GNNs

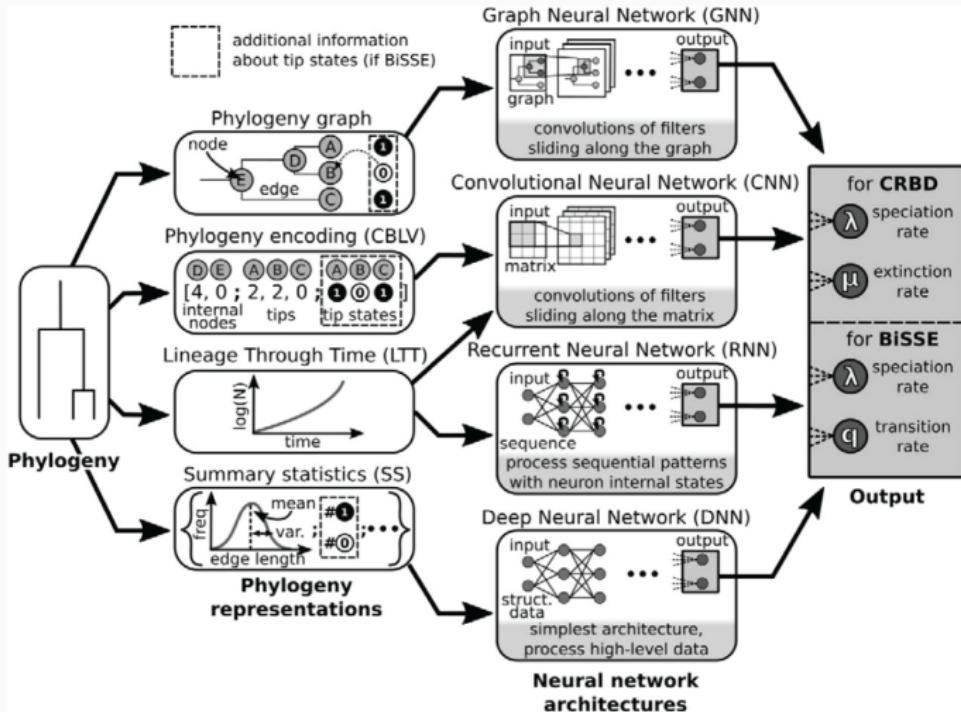


Figure 1 from Latjaaiti et al., 2023

A motivating example for GNNs

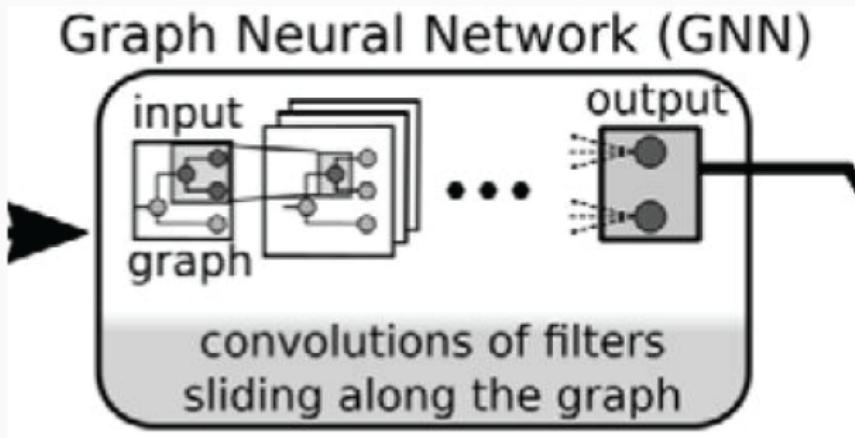


Figure 1 from Latjaaiti et al., 2023

A motivating example for GNNs

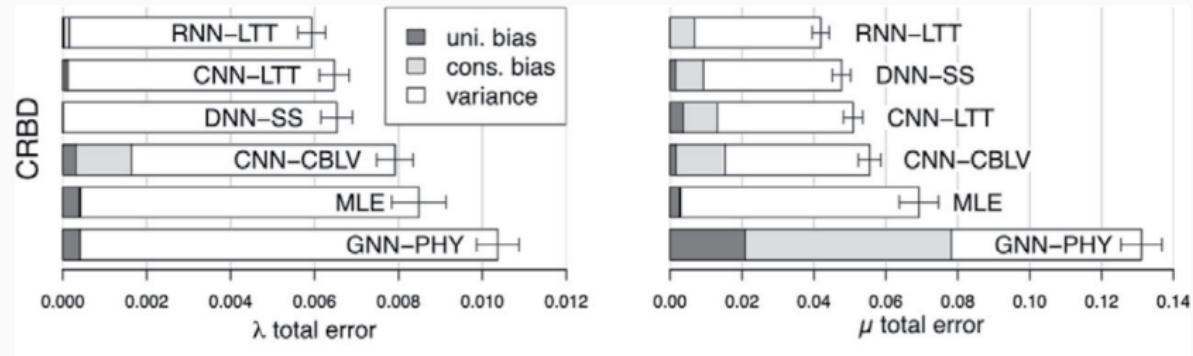


Figure 3 from Latjaaiti et al., 2023

- GNNs take graphs as input!
- GNNs preserve important aspects of graph architecture.
- Even though they didn't work well for this problem, I'm very excited about how GNNs might be used in phylogenetics.

Outline

Overview of Supervised Machine Learning Algorithms

Decision Trees

Fully Connected Neural Networks (FCNNs)

Convolutional Neural Networks (CNNs)

Graphical Neural Networks

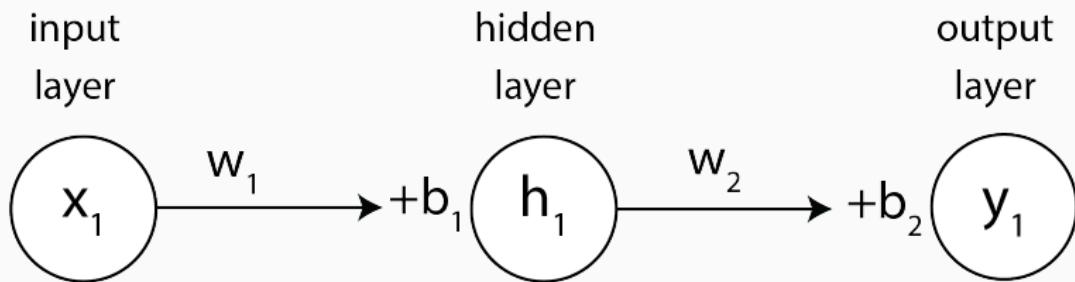
Recurrent Neural Networks

Generative Models

Overview of Algorithms

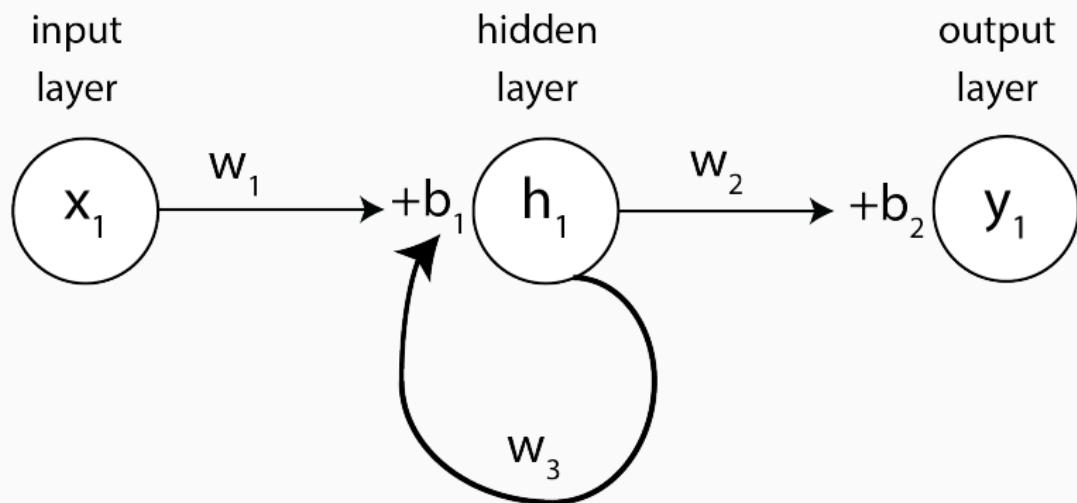
What is a RNN?

My dog, Dexter, loves to chase squirrels.



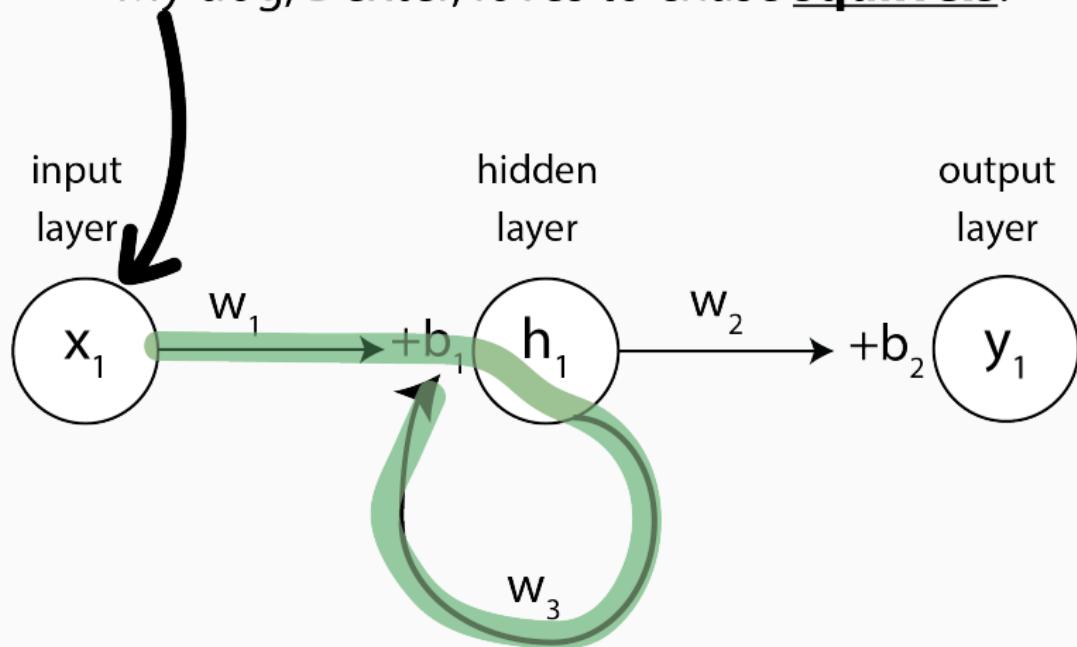
What is a RNN?

My dog, Dexter, loves to chase squirrels.



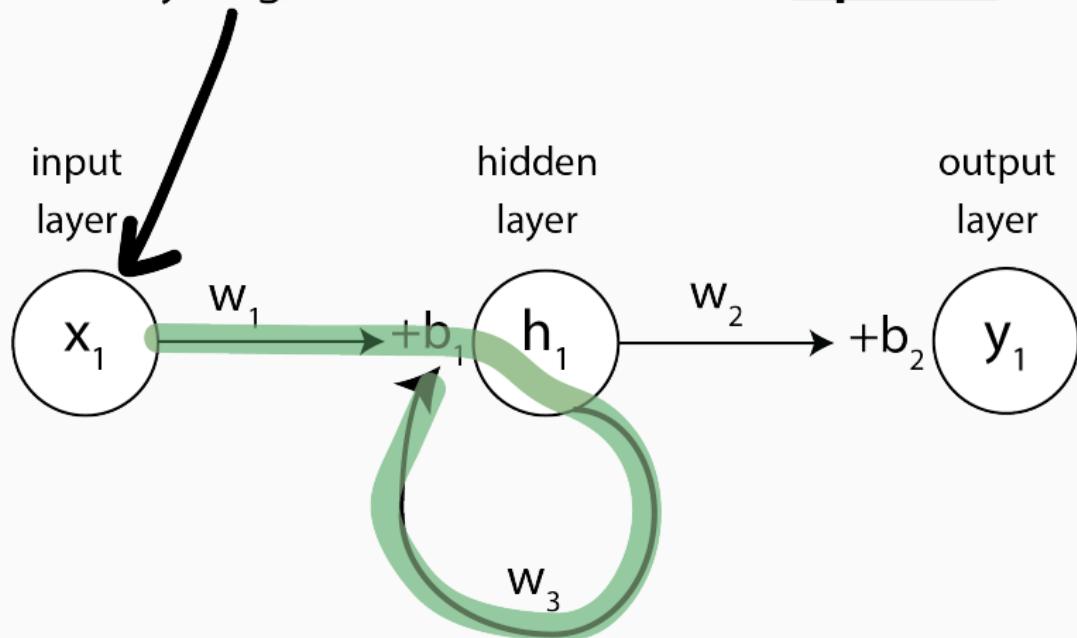
What is a RNN?

My dog, Dexter, loves to chase squirrels.



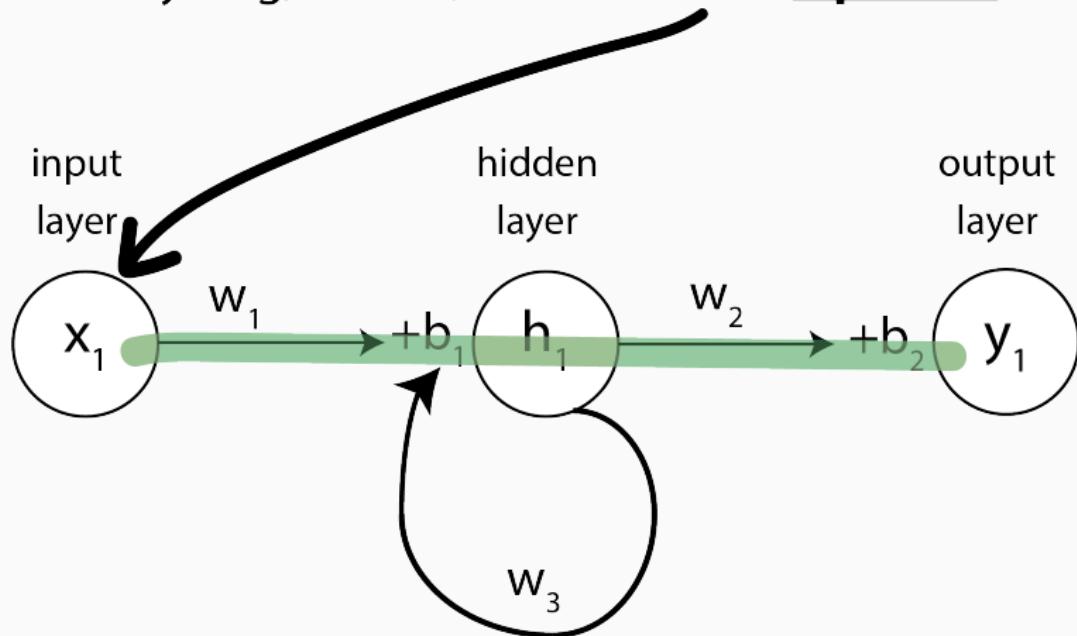
What is a RNN?

My dog, Dexter, loves to chase squirrels.



What is a RNN?

My dog, Dexter, loves to chase squirrels.



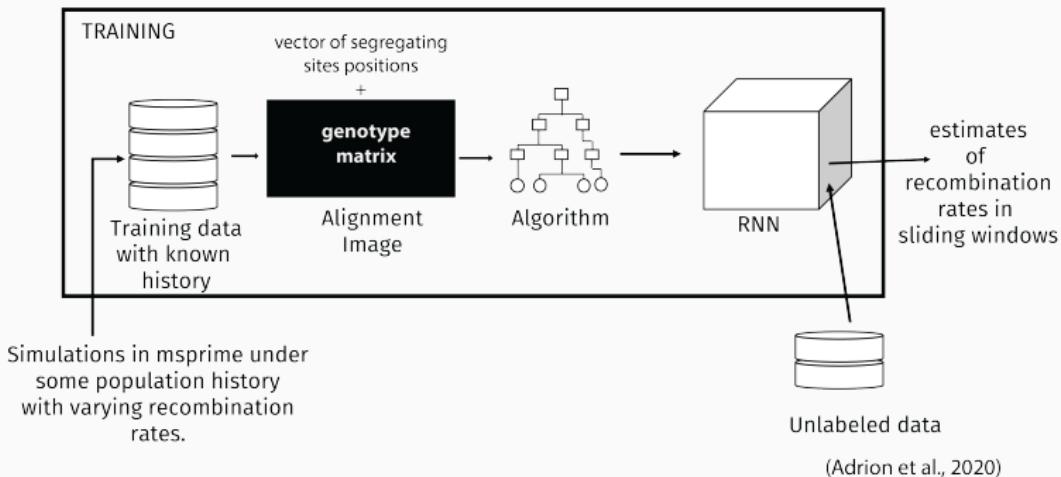
What is a RNN?

- Can consider different amounts of sequential data.
- Takes advantage of correlations between sequential data using feedback loops.

Why might we use a RNN in population genetics?

- Some properties of the genome are spatially autocorrelated.
- For example, we might expect that recombination rates are more similar between nearby regions than between dispersed regions of the genome.
- Adrián et al., (2020) were motivated by this to develop ReLERNN, a RNN for predicting recombination rate across the genome.
- ReLERNN outperformed the CNN developed by Flagel et al. (2018) to estimate recombination rates for genomic windows.

An example: ReLERNN



An example: ReLERNN

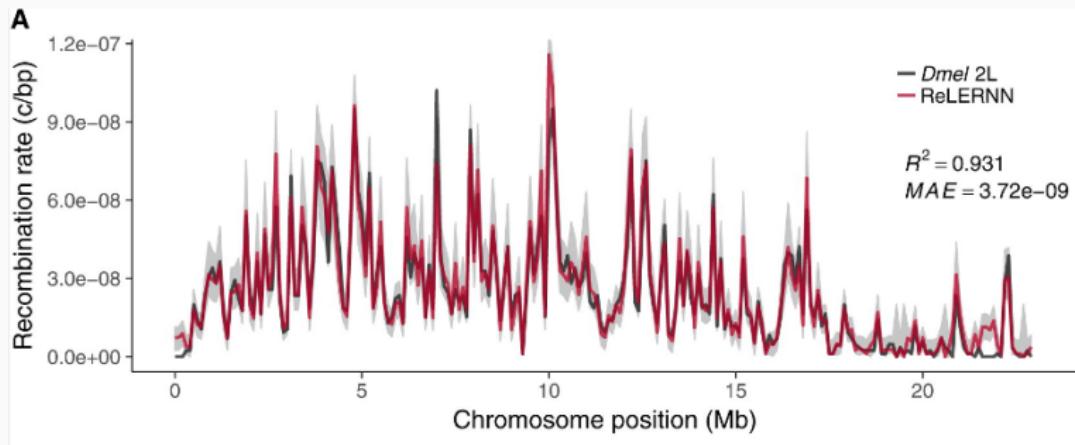


Figure 2 from Adrion et al., 2020

Outline

Overview of Supervised Machine Learning Algorithms

Decision Trees

Fully Connected Neural Networks (FCNNs)

Convolutional Neural Networks (CNNs)

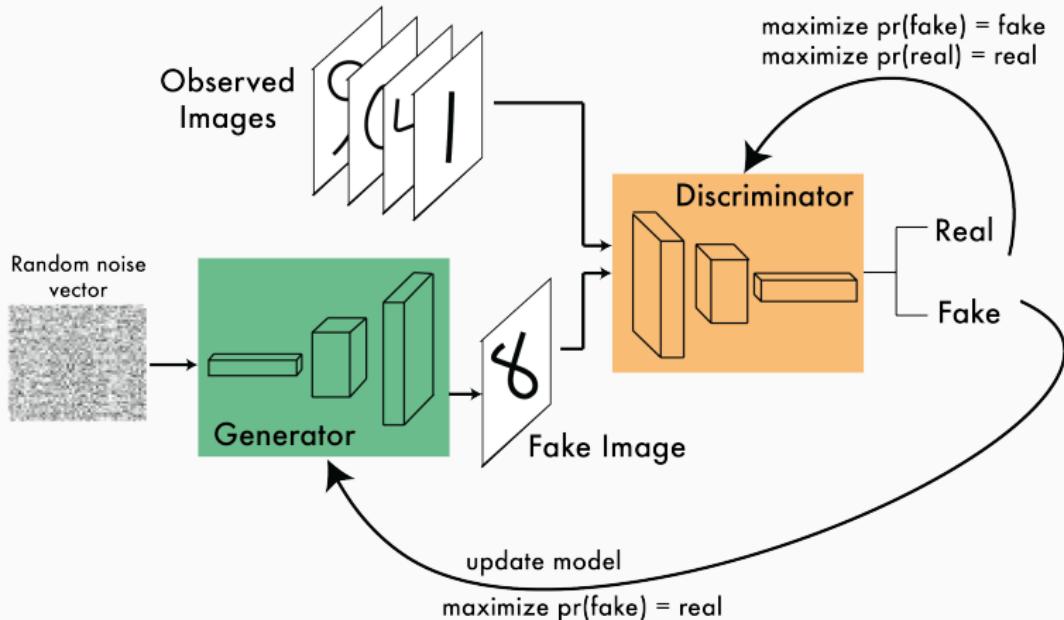
Graphical Neural Networks

Recurrent Neural Networks

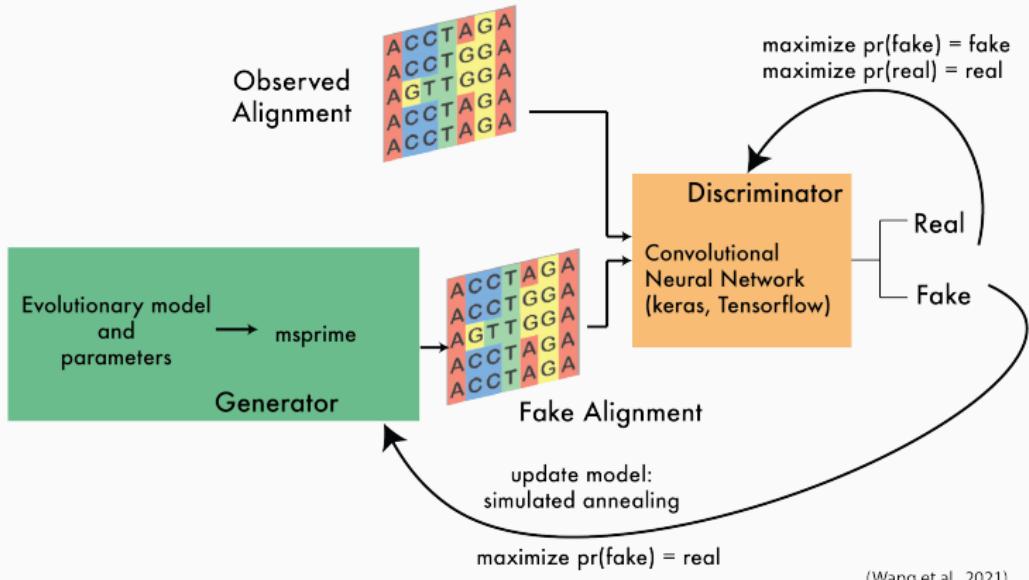
Generative Models

Overview of Algorithms

What is a Generative Adversarial Network?



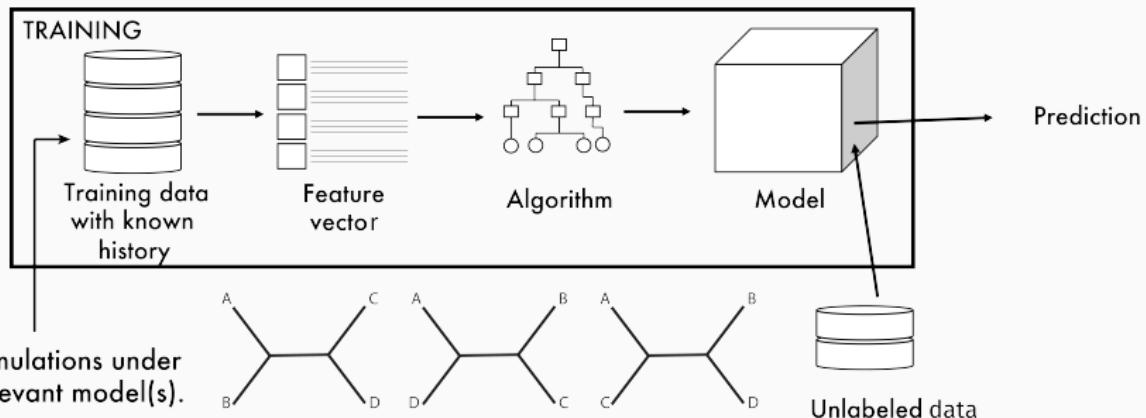
How have GANs been used in population genetics?



(Wang et al., 2021)

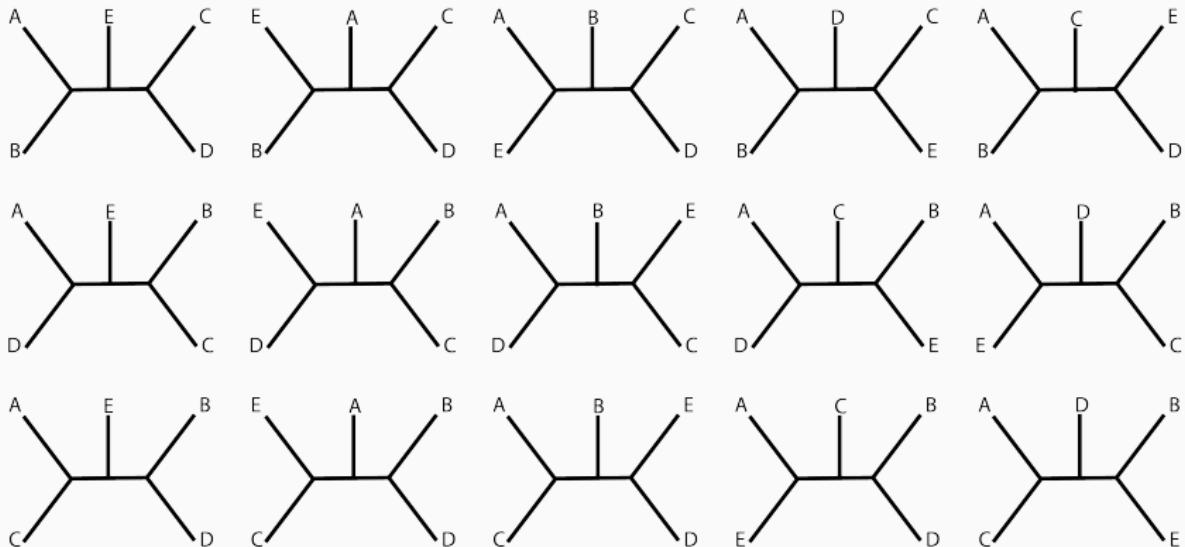
Why might we need a GAN for phylogenetics?

Traditional Machine Learning approaches require that simulations are conducted under all relevant models prior to training.



Why might we need a GAN for phylogenetics?

Five taxa: Fifteen unrooted trees.

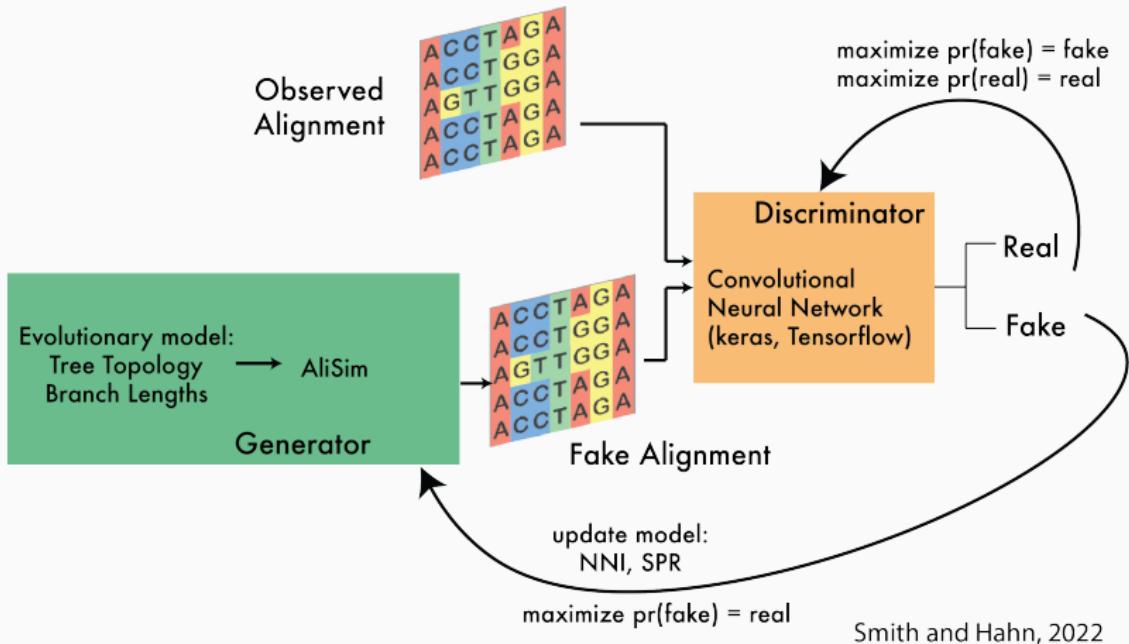


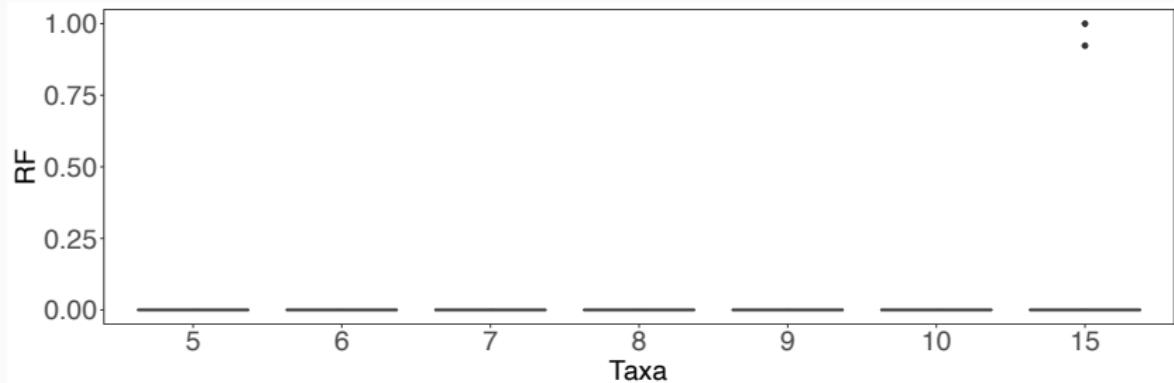
Why might we need a GAN for phylogenetics?

In phylogenetics, the model space is (at a minimum) the tree space.
The number of possible trees becomes prohibitively large even with
moderate numbers of taxa.

TAXA	TREES
3	1
4	3
5	15
6	105
7	954
8	10,395
9	135,135
10	2,027,025

phyloGAN





Challenges with GANs

- GANs can be very difficult to train stably.
- GANs rely on similarities between empirical and generated data, which can be a problem when our models are not well-specified.

Outline

Overview of Supervised Machine Learning Algorithms

Decision Trees

Fully Connected Neural Networks (FCNNs)

Convolutional Neural Networks (CNNs)

Graphical Neural Networks

Recurrent Neural Networks

Generative Models

Overview of Algorithms

Random Forests

- Random Forest Classifiers (and Regressors) use collections of decision trees trained for the task at hand.
- Hyperparameters include the number of trees, the number of features considered when splitting a node, and parameters controlling the size of each decision tree.
- Random Forests rely on hand-crafted summary statistics.
- Examples using RF include delimitR and FILET.

Fully Connected Neural Networks

- Fully connected neural networks consist of dependent non-linear functions.
- Layers contain nodes, which are connected with weights and associated with biases. These weights and biases are the trainable parameters of the network.
- Hyperparameters include the number of hidden layers, the number of neurons per layer, the activation functions and the batch sizes and number of epochs used in training.
- FCNNs usually rely on hand-crafted summary statistics.
- Examples include the evoNET network introduced by Sheehan and Song (2016) and ml4ils (Rosenzweig et al., 2022).

Convolutional Neural Networks

- CNNs are very similar to FCNNs, but at the beginning we perform a series of convolutions, which allows us to process image data in a meaningful and efficient way.
- Trainable parameters include the weights and biases associated with the filters (plus weights and biases associated with the FCNN).
- In addition to the hyperparameters of the FCNN, hyperparameters include the number of convolutional layers, the size and shape of filters, the number of filters, the stride, and the type of pooling to use.
- CNNs can directly process images of alignment, bypassing the need to calculate summary statistics.
- We discussed an approach for detecting introgressed genomic windows (Flagel et al., 2018), and an approach to infer quartet trees (Suvorov et al., 2020).

- GNNs are similar to FCNNs in many ways, but the input is a graph!
- We perform convolutions on the graph in a way that preserves aspects of graph structure.
- GNNs are an obvious architecture for analyzing phylogenies, but it may take some work to figure out how to best use them for this purpose.

Recurrent Neural Networks

- Recurrent Neural Networks extend FCNNs to handle sequential input data.
- The hyperparameters are similar to those of a FCNN.
- We discussed ReLERNN, a method for estimating recombination rates (Adrion et al., 2020).

Generative Adversarial Networks

- GANs consist of a generator, which tries to produce a realistic data, and a discriminator, which tries to distinguish generated from real data.
- By using an evolutionary model and parameters as the generator, we can make evolutionary inferences.
- Examples include pg-GAN (Wang et al., 2021) and phyloGAN (Smith and Hahn, 2022).

Challenges and Future Directions

Outline

Challenges and Future Directions

Overfitting

Hyperparameter tuning

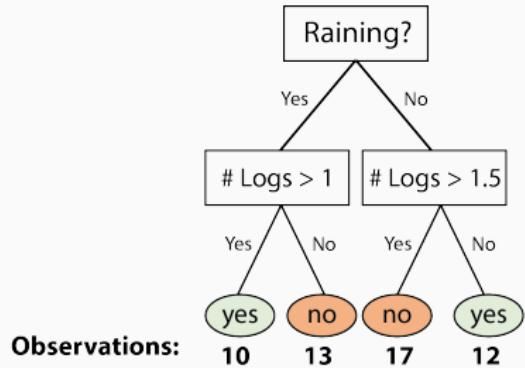
Simulation Misspecification

The Black Box

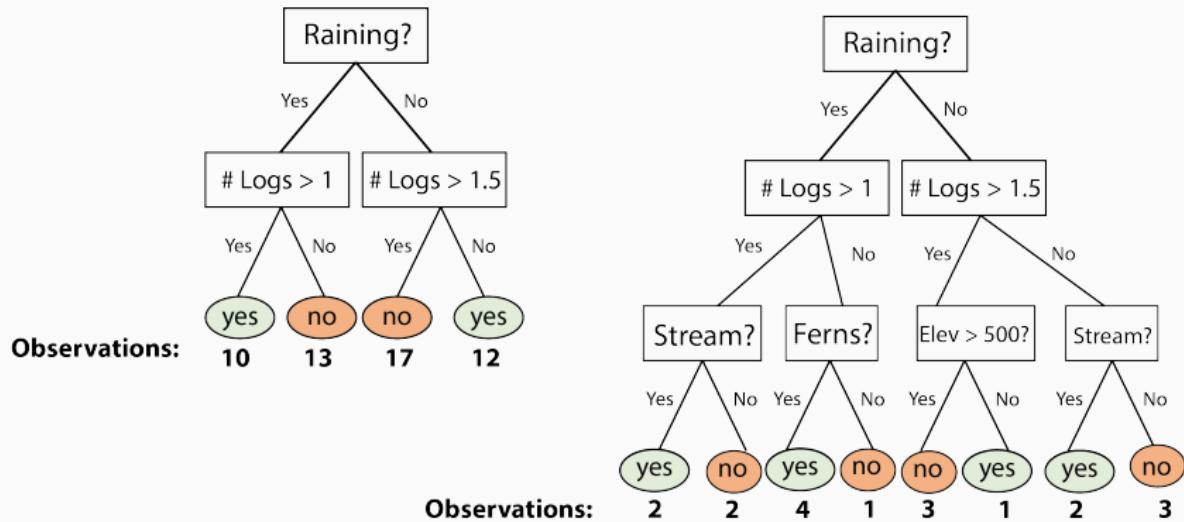
How do I avoid overfitting?

- Overfitting occurs when the model gives accurate predictions for the training data, but not new data.
- We can attempt to avoid overfitting using several approaches:

Overfitting: Random Forests



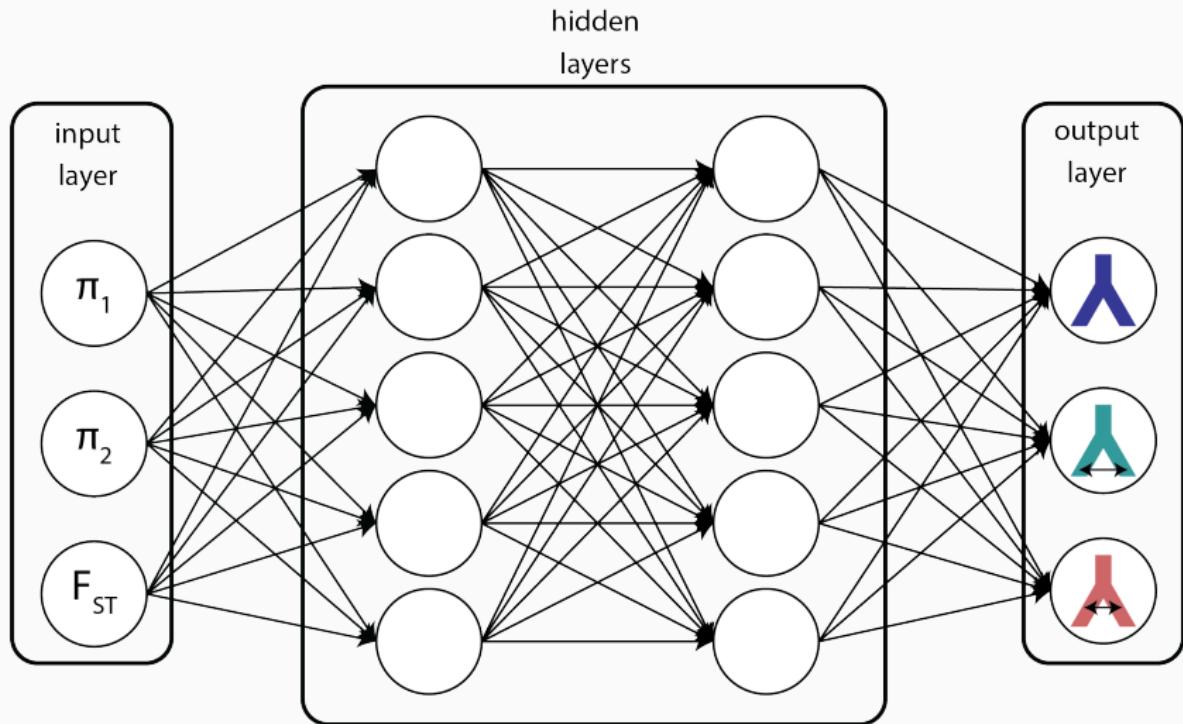
Overfitting: Random Forests



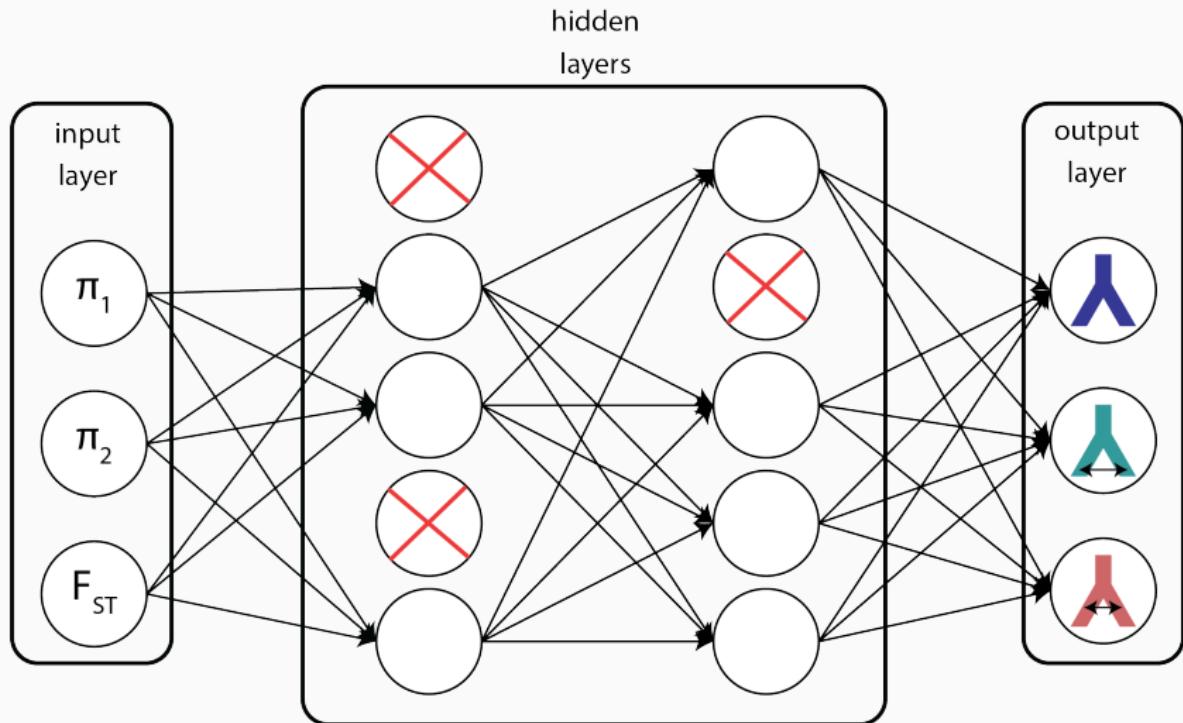
How do I avoid overfitting?

- Overfitting occurs when the model gives accurate predictions for the training data, but not new data.
- We can attempt to avoid overfitting using several approaches:
 1. RF: don't let decision trees get too large

Overfitting: Neural Networks



Overfitting: Neural Networks



How do I avoid overfitting?

- Overfitting occurs when the model gives accurate predictions for the training data, but not new data.
- We can attempt to avoid overfitting using several approaches:
 1. RF: don't let decision trees get too large
 2. NN: dropout layers

How do I avoid overfitting?

- Overfitting occurs when the model gives accurate predictions for the training data, but not new data.
- We can attempt to avoid overfitting using several approaches:
 1. RF: don't let decision trees get too large
 2. NN: dropout layers
 3. NN: use regularization (e.g., L1 regularization)

How do I avoid overfitting?

- Overfitting occurs when the model gives accurate predictions for the training data, but not new data.
- We can attempt to avoid overfitting using several approaches:
 1. RF: don't let decision trees get too large
 2. NN: dropout layers
 3. NN: use regularization (e.g., L1 regularization)
 4. NN: use early stopping

How do I avoid overfitting?

- Overfitting occurs when the model gives accurate predictions for the training data, but not new data.
- We can attempt to avoid overfitting using several approaches:
 1. RF: don't let decision trees get too large
 2. NN: dropout layers
 3. NN: use regularization (e.g., L1 regularization)
 4. NN: use early stopping
 5. NN: reduce model complexity

How do I avoid overfitting?

- Overfitting occurs when the model gives accurate predictions for the training data, but not new data.
- We can attempt to avoid overfitting using several approaches:
 1. RF: don't let decision trees get too large
 2. NN: dropout layers
 3. NN: use regularization (e.g., L1 regularization)
 4. NN: use early stopping
 5. NN: reduce model complexity
- To ensure you recognize overfitting, be sure to keep an independent test set!

Outline

Challenges and Future Directions

Overfitting

Hyperparameter tuning

Simulation Misspecification

The Black Box

How do I choose my hyperparameters

Hyperparameters can be optimized using several approaches:

1. Manual optimization
2. Grid search
3. Random search

Outline

Challenges and Future Directions

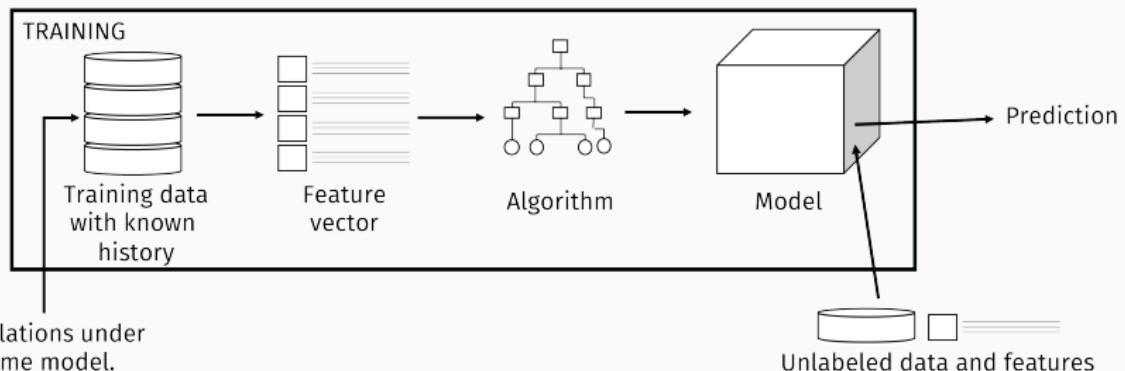
Overfitting

Hyperparameter tuning

Simulation Misspecification

The Black Box

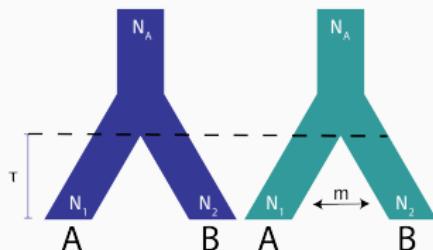
Simulation Misspecification



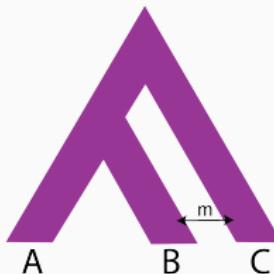
Simulation Misspecification

What if our simulations aren't realistic?

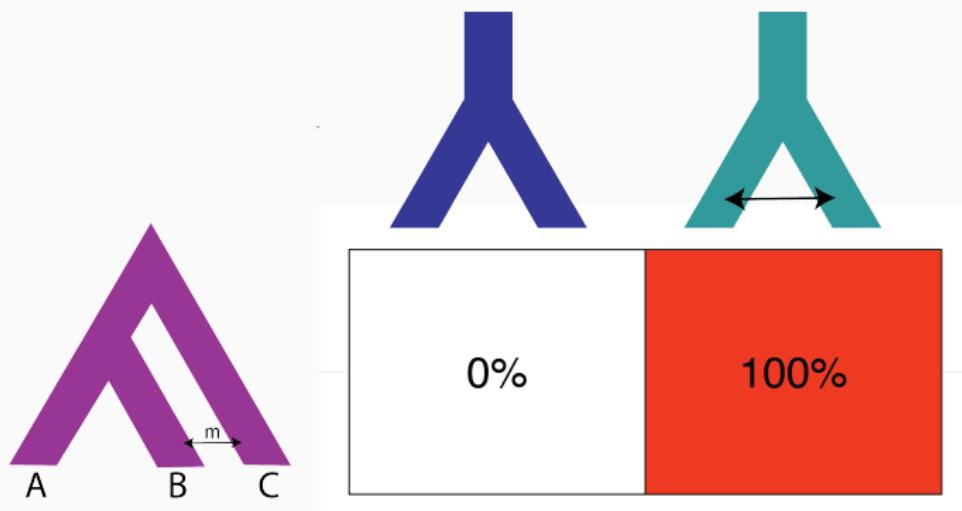
Simulations:



A more realistic model:



Simulation Misspecification

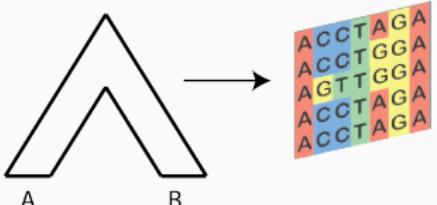
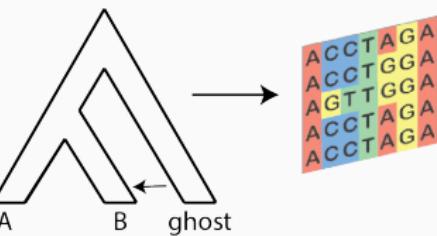


Domain Shifts

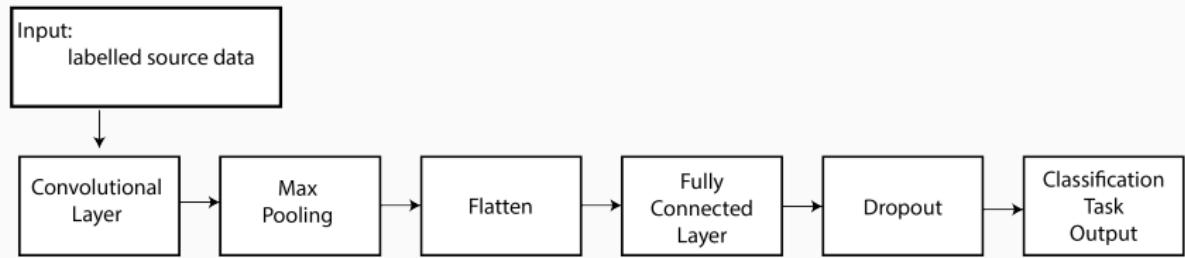
Domain shifts occur when training data and test data arise from different distributions.

Domain adaptation aims to build networks that perform well when the test data comes from a different distribution than the training data.

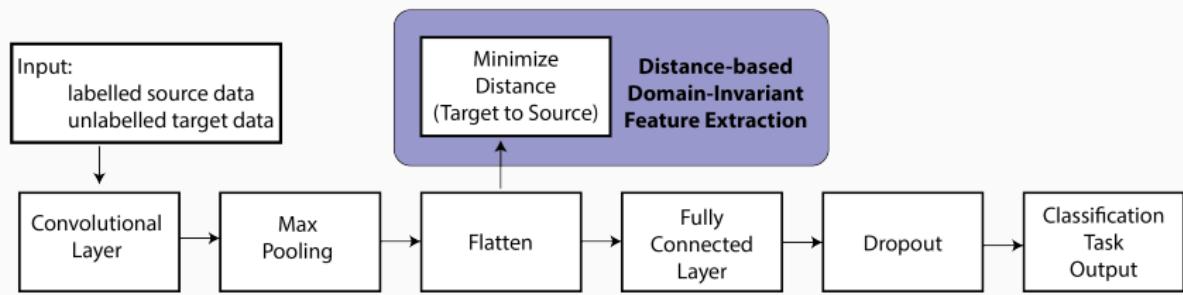
Domain Shifts

	Domain shift in digit classification	Domain shift in population genetics
Source data		
Target data		

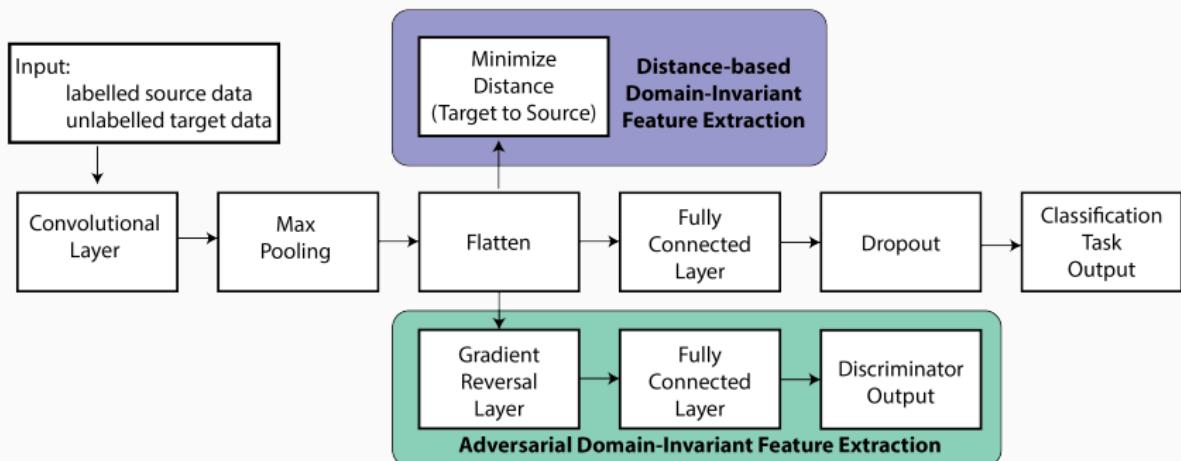
Domain Adaptation



Domain Adaptation



Domain Adaptation

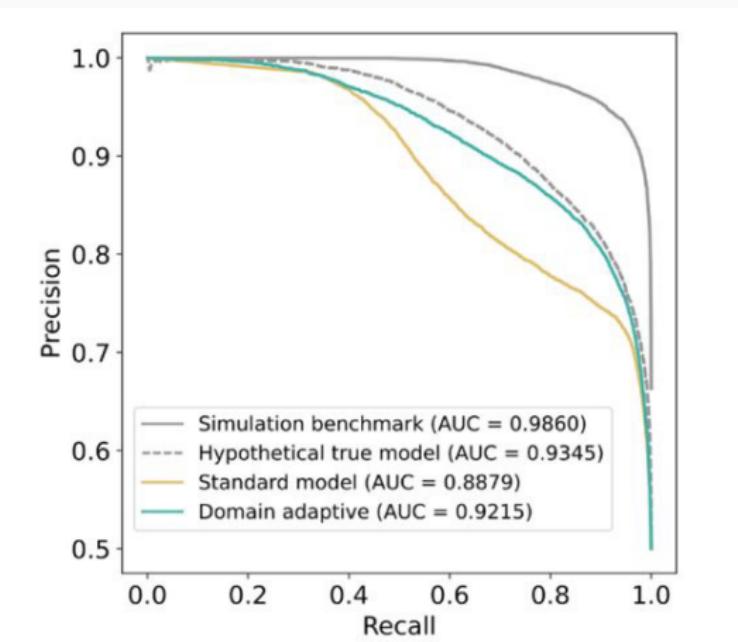


Domain Adaptation

Mo and Siepel (2023) used domain adaptation.

Goal: to estimate the selection coefficient

Misspecification: Background Selection



Outline

Challenges and Future Directions

Overfitting

Hyperparameter tuning

Simulation Misspecification

The Black Box

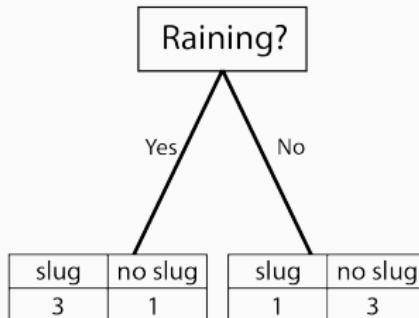
The Black Box

Can we understand how features are driving predictions?

Variable Importance

Will I find a slug?

Raining?	Stream?	# Logs	Slug?
yes	no	2	yes
yes	yes	2	yes
yes	yes	4	yes
no	yes	1	yes
yes	no	0	no
no	yes	1	no
no	yes	2	no
no	no	2	no



GINI Index

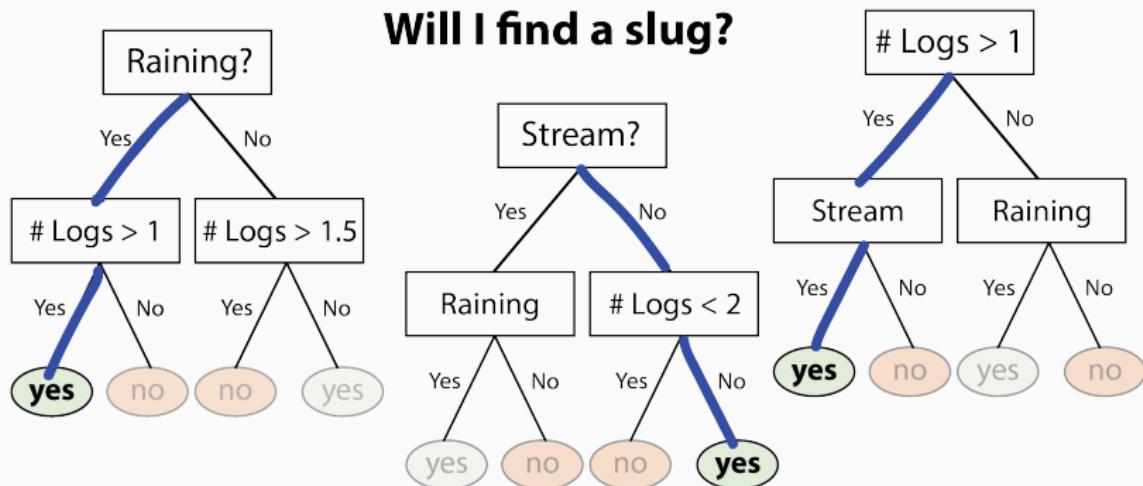
measure of node purity

0: perfectly pure

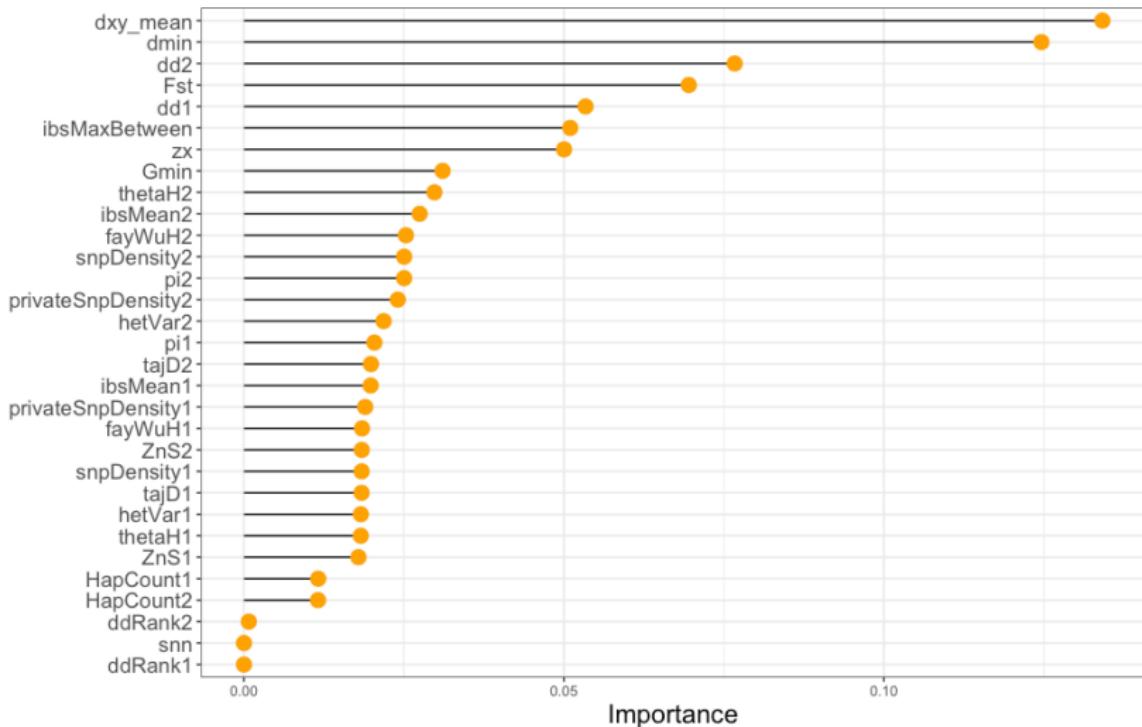
1: elements randomly distributed

$$GINI = 1 - \sum_{i=1}^n p_i^2$$

Variable Importance in Random Forests



Variable Importance in Random Forests



Variable importance from Schrider et al., 2018

Variable Importance in Neural Networks

Permutation testing!

- For each variable i :
 1. Randomly permute the values of the variable across the test datasets.
 2. Apply the neural network.
 3. Measure the increase in prediction error relative to the baseline.

Variable Importance in Neural Networks

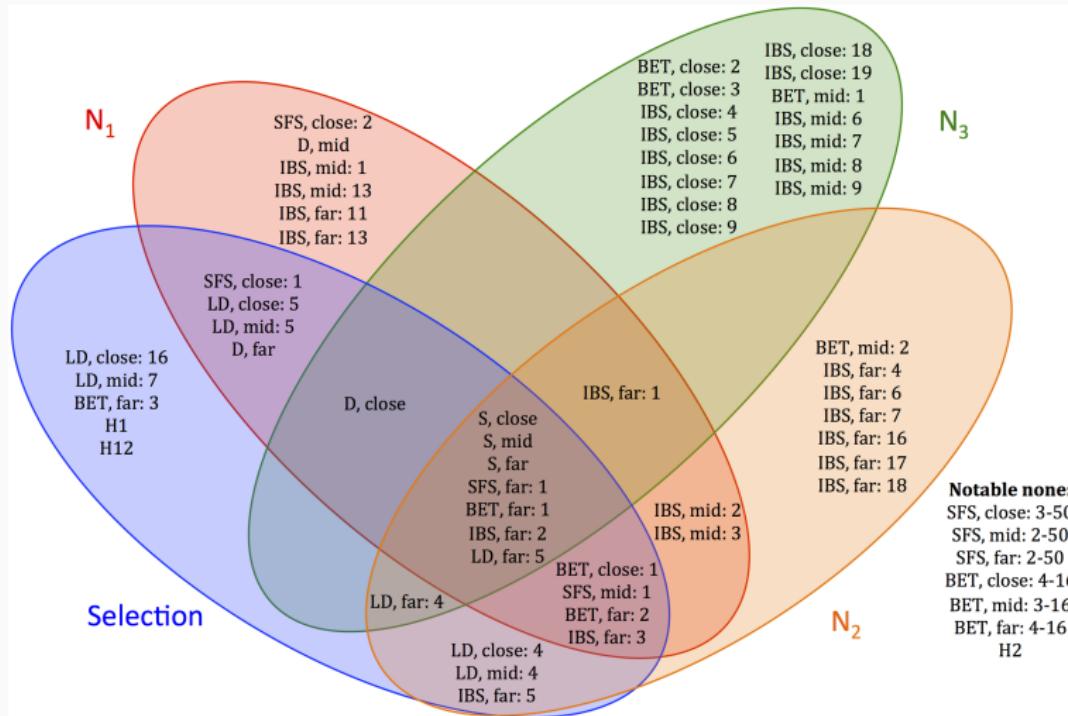
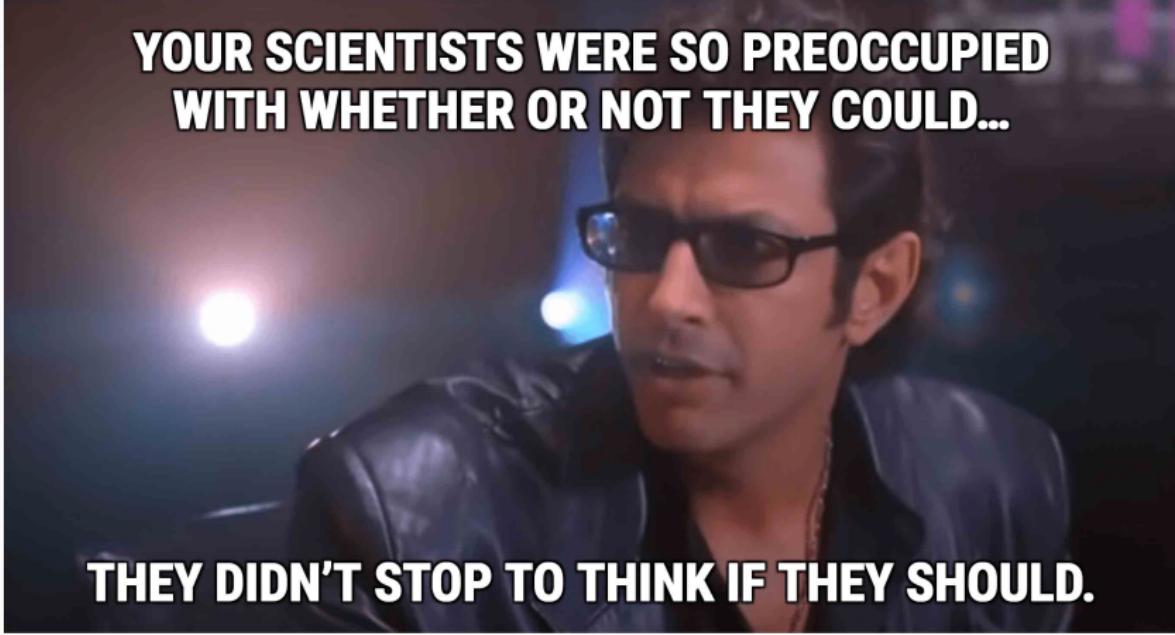


Figure 5 from Sheehan and Song, 2016

Does this really solve the "Black Box" problem?



**YOUR SCIENTISTS WERE SO PREOCCUPIED
WITH WHETHER OR NOT THEY COULD...**

THEY DIDN'T STOP TO THINK IF THEY SHOULD.

Does this really solve the "Black Box" problem?

- Selection can mislead inferences of migration and population expansion.
- Traditional methods do not consider selection.
- I can build a machine learning algorithm that infers migration rates or population size changes through time!

Does this really solve the "Black Box" problem?

- That machine learning approach may be extremely effective on simulated data.
- It will give me an answer on my real data!
- BUT, it's important to carefully consider other processes that could produce the results, whether my model may be overfit to my simulated data, and whether my results make sense in the light of other analyses!

How and when should I use
Machine Learning?

How and when to use machine learning

1. Can I achieve my goal using a full Likelihood or Bayesian approach?
2. What do I want to predict (i.e., what would the output of my network look like)?
3. What kind of data do I have, and how can I best structure that data to answer the task at hand?
4. What processes do I need to account for when generating my training data, and which simulator can best accomplish this task?
5. Which process am I ignoring that might impact inference?
6. What computational resources can I use?

Useful Tools

A (non-exhaustive) list of useful simulators

1. msprime (Baumdicker et al., 2022): integrates easily into python workflows
2. Sim-Phy (Mallo et al., 2015): great for simulating gene duplication and loss
3. SLiM (Haller and Messer, 2019): forward-in-time simulations with selection
4. Ali-Sim (Ly-Trong et al., 2022): great for phylogenetics

Implementing Machine Learning workflows

1. tensorflow
2. Sci-kit learn
3. PyTorch

Jupyter Notebook Example
