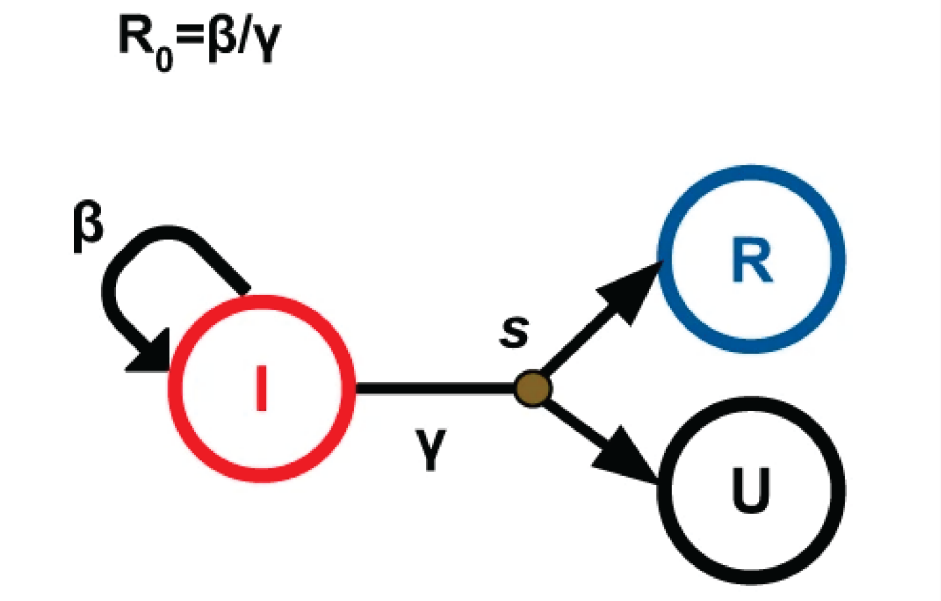
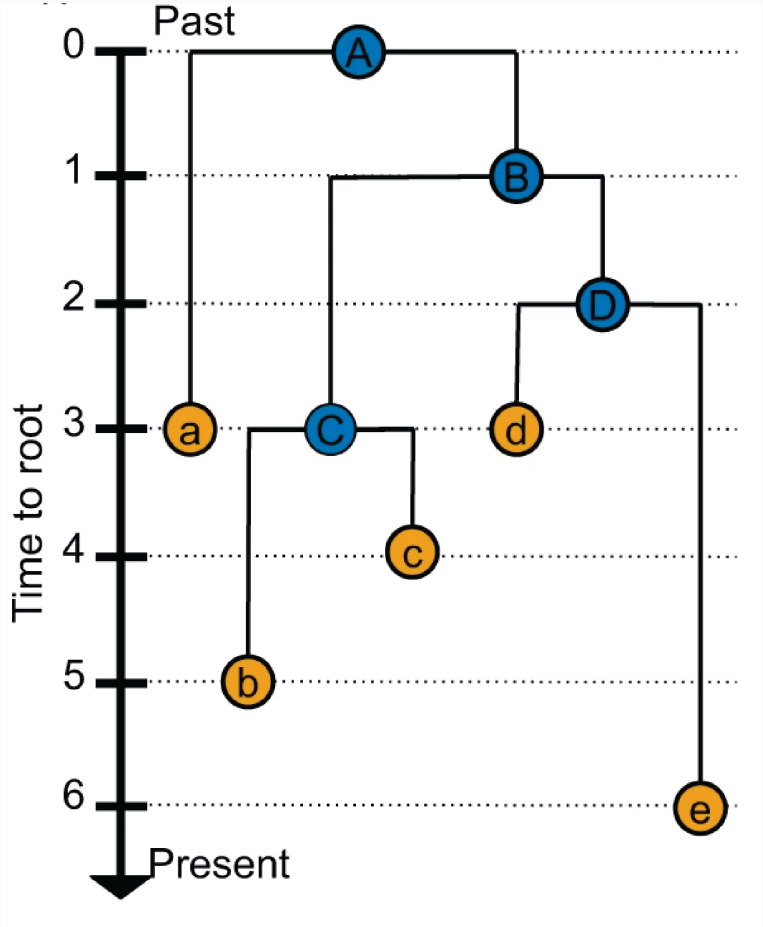
Machine learning in population and phylogenetics

Megan L Smith
Woods Hole Workshop on Molecular Evolution

Outline

A motivating example for GNNs



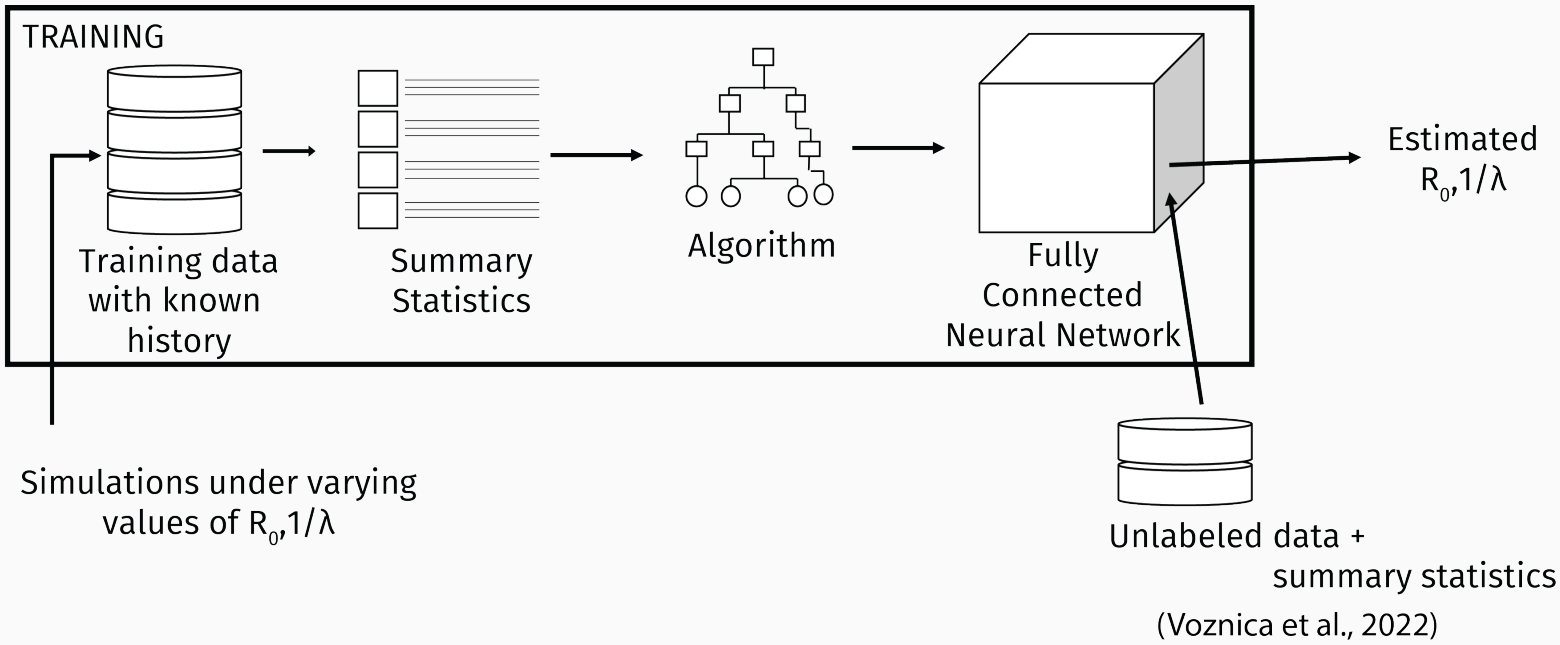


Figures 1, 2 from Voznica et al., 2022

A motivating example for GNNs

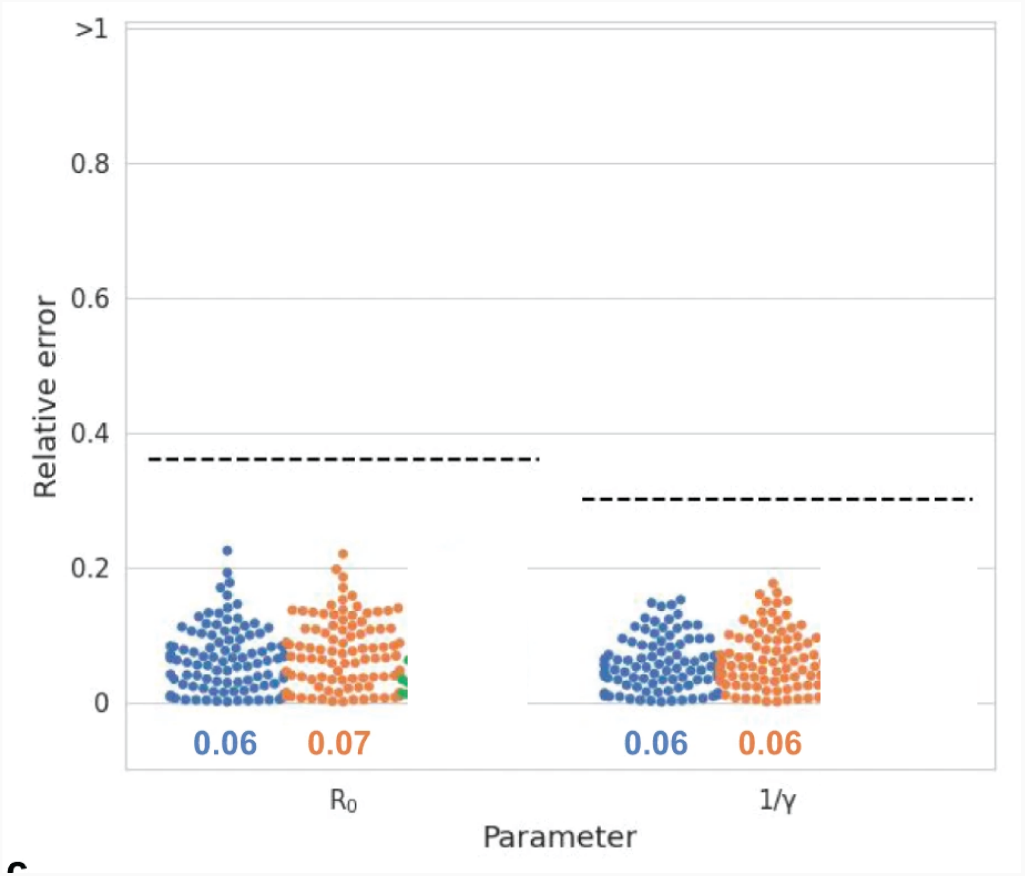
- How could you use machine learning to estimate these parameters?
- Available data:
    1. sequence data
    2. phylogeny

A motivating example for GNNs



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

A motivating example for GNNs



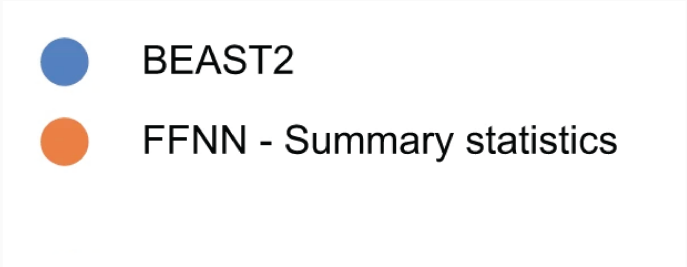


Figure 3 from Voznica et al., 2022

A motivating example for GNNs

Compact Bijective Ladderized Vector encoding

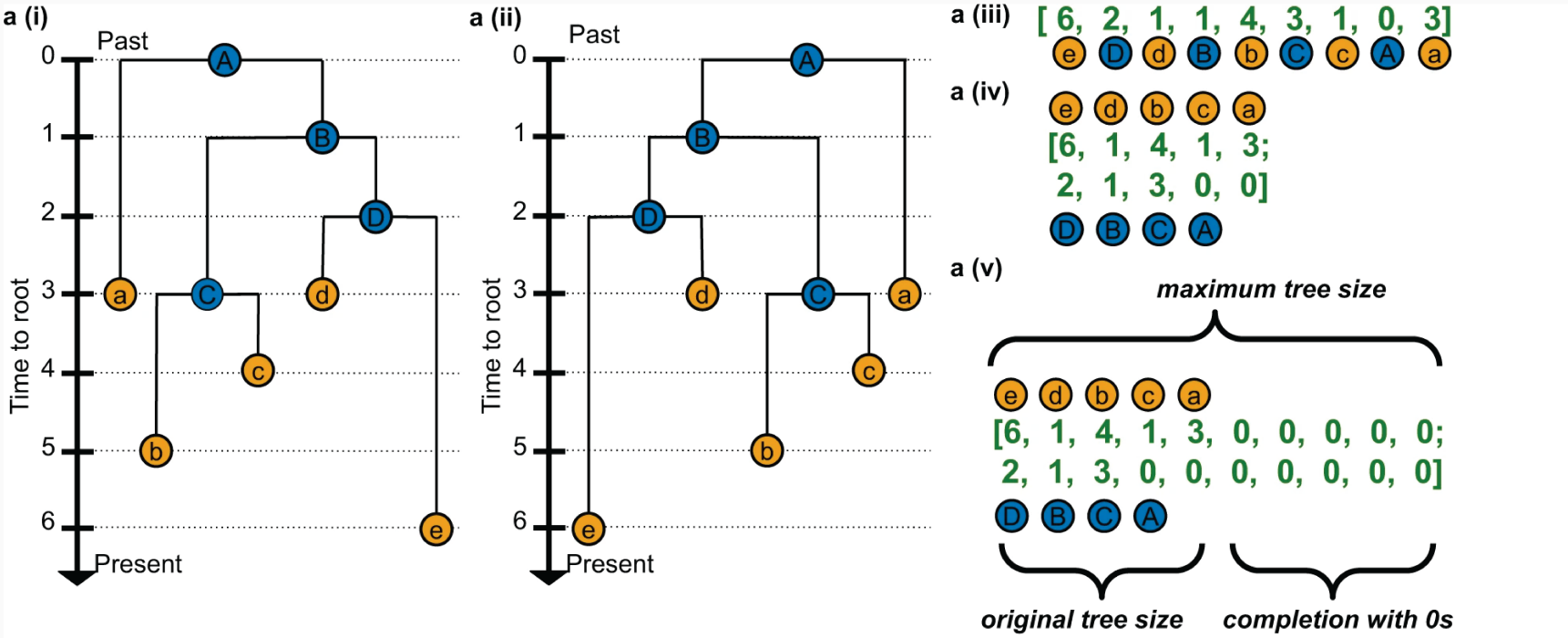
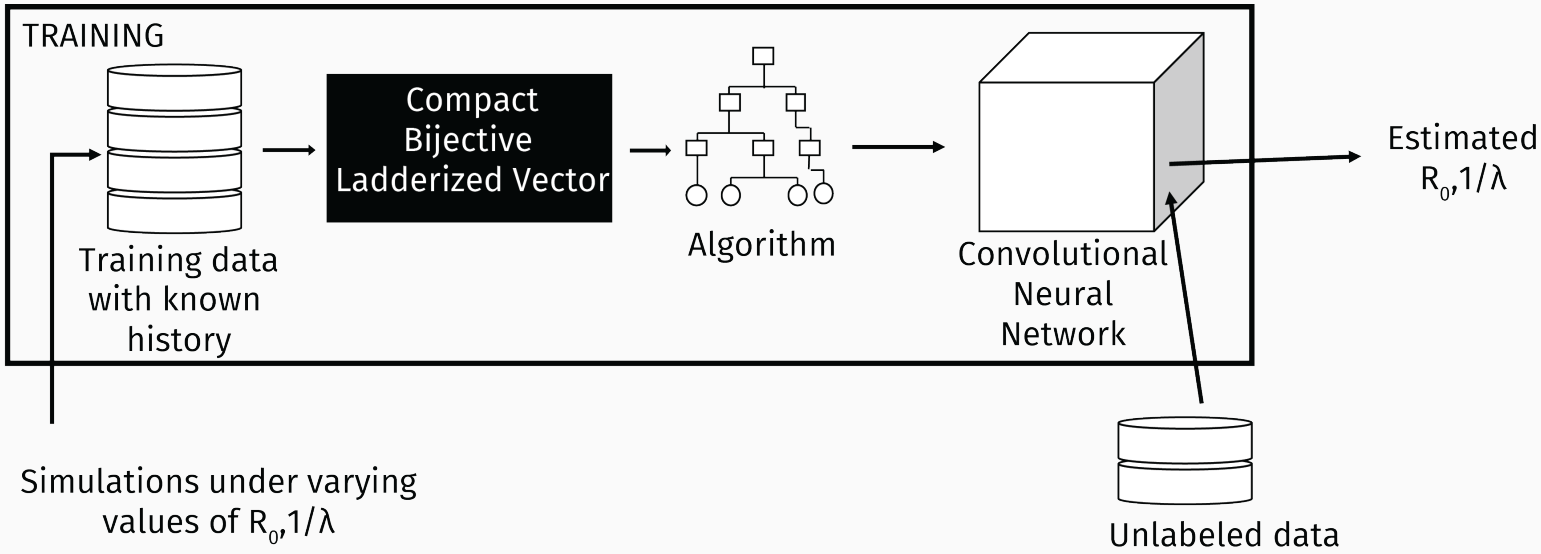


Figure 2 from Voznica et al., 2022

A motivating example for GNNs



(Voznica et al., 2022)

A motivating example for GNNs

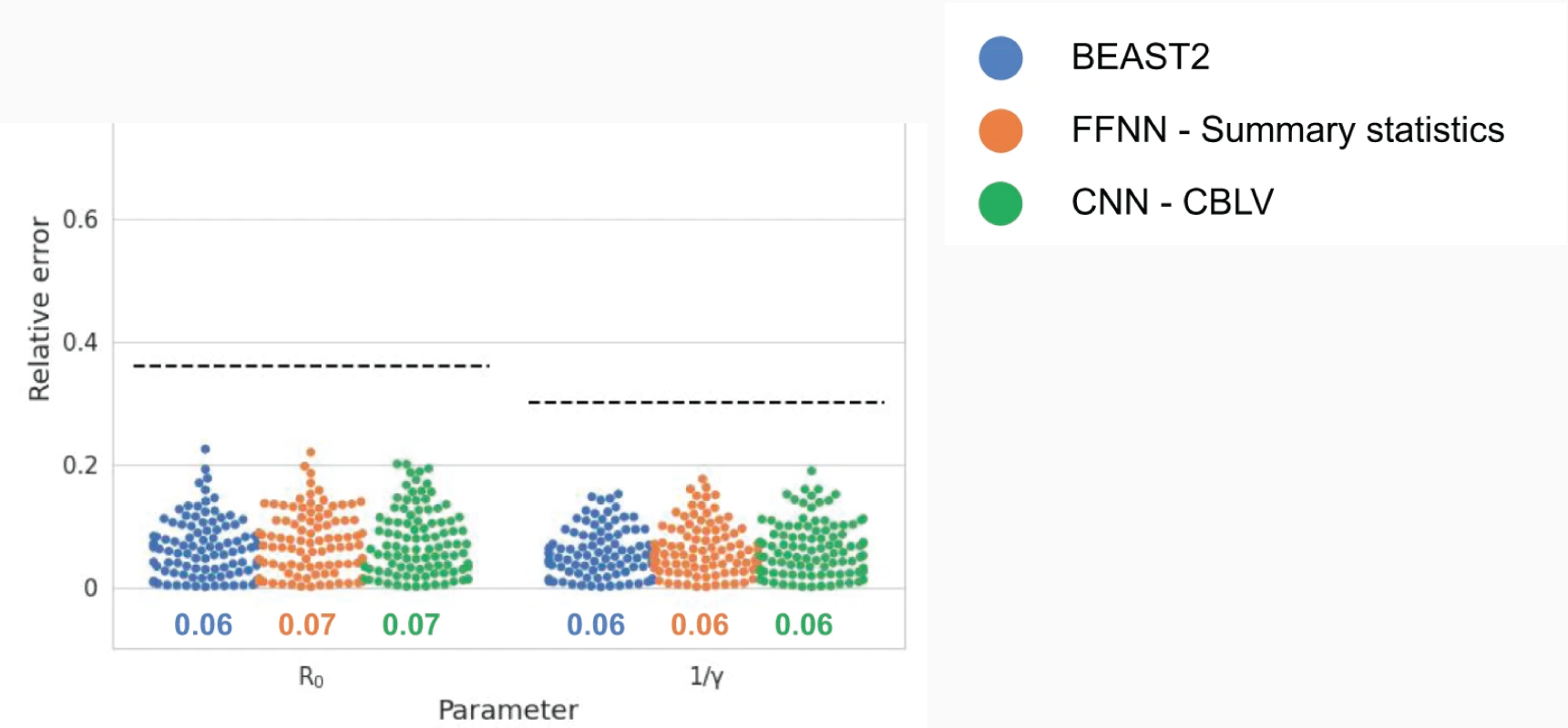


Figure 3 from Voznica et al., 2022

A motivating example for GNNs

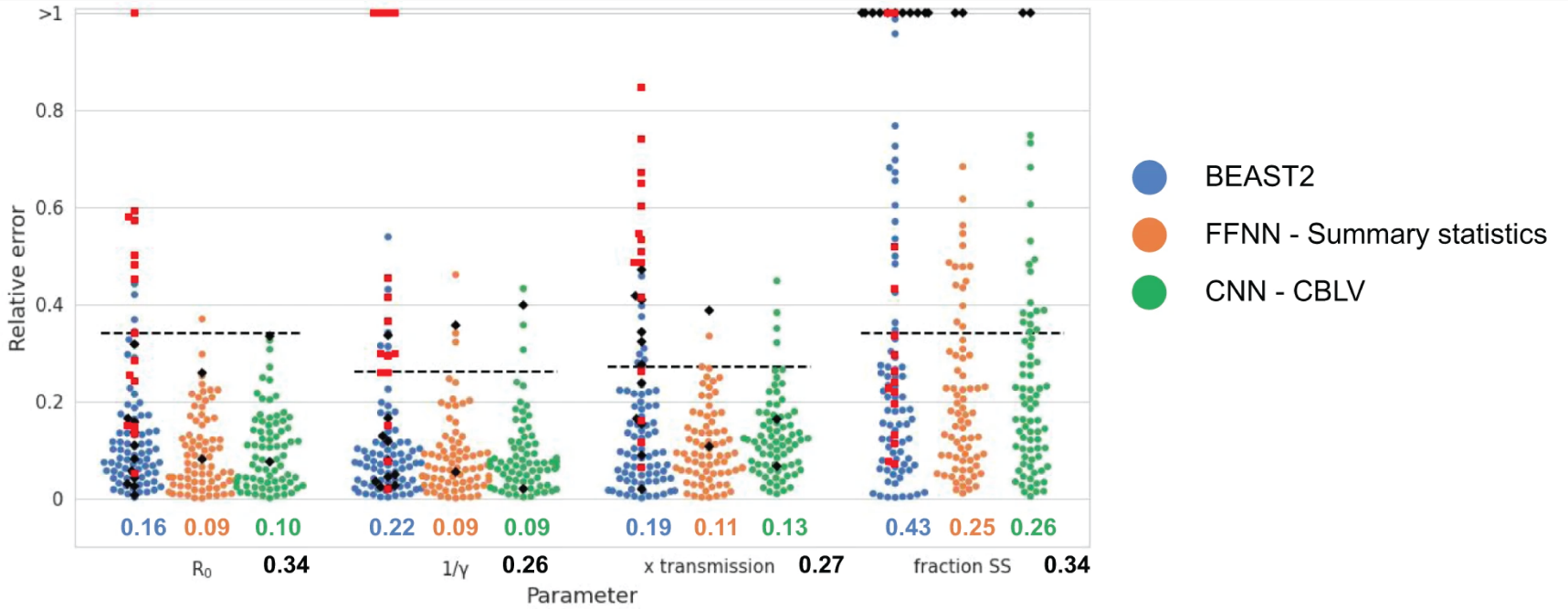


Figure 3 from Voznica et al., 2022

A motivating example for GNNs

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

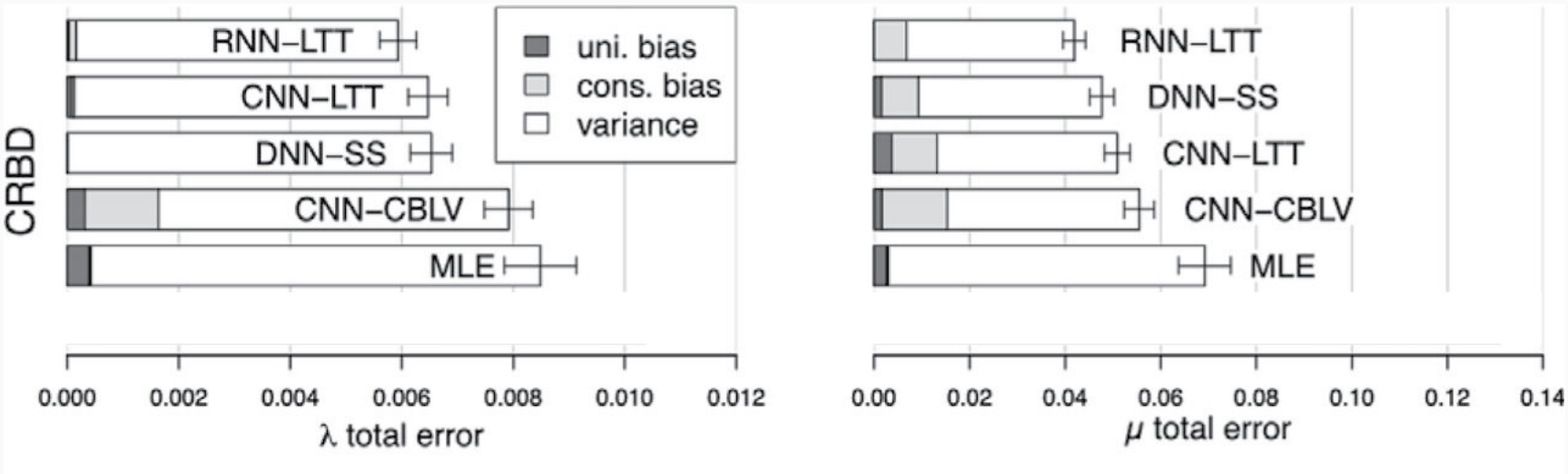
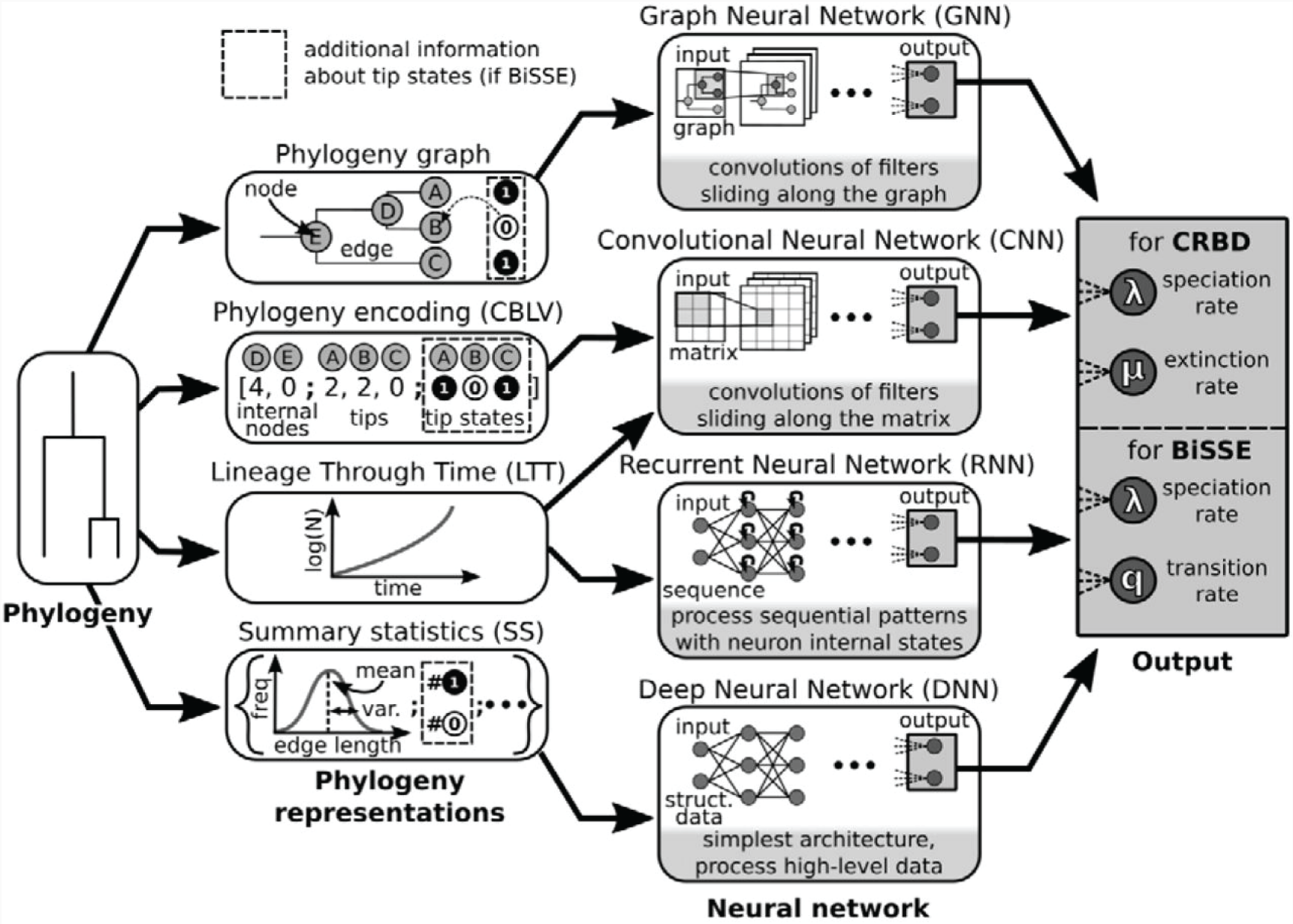


Figure 3 from Latjaaiti et al., 2023

A motivating example for GNNs



A motivating example for GNNs

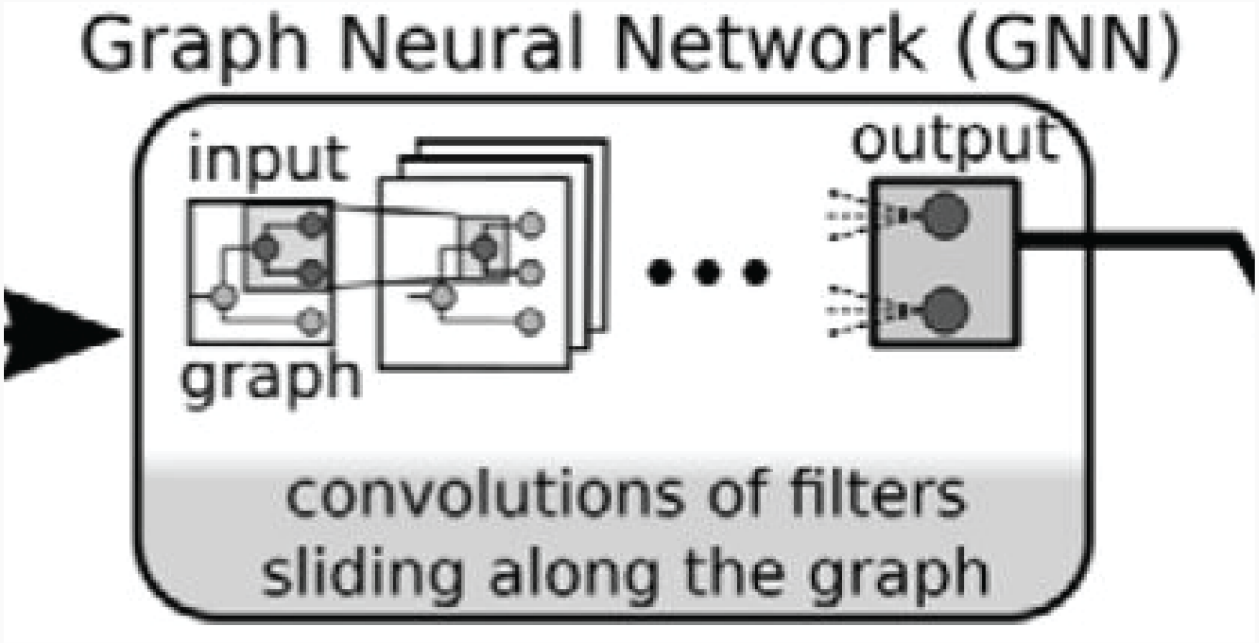


Figure 1 from Latjaaiti et al., 2023

A motivating example for GNNs

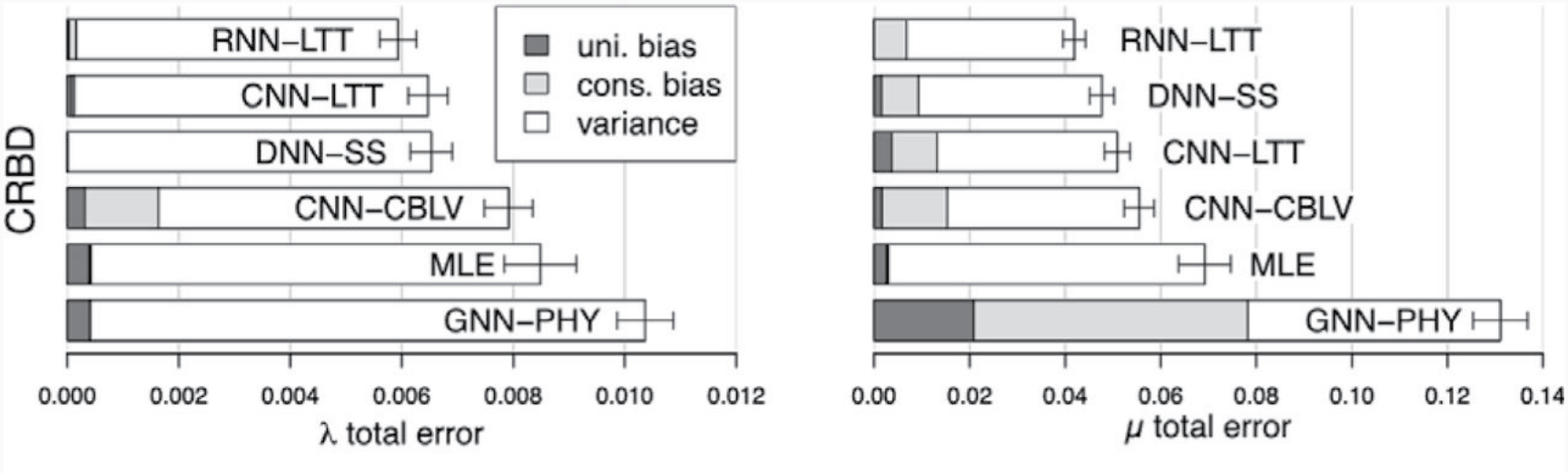


Figure 3 from Latjaaiti et al., 2023

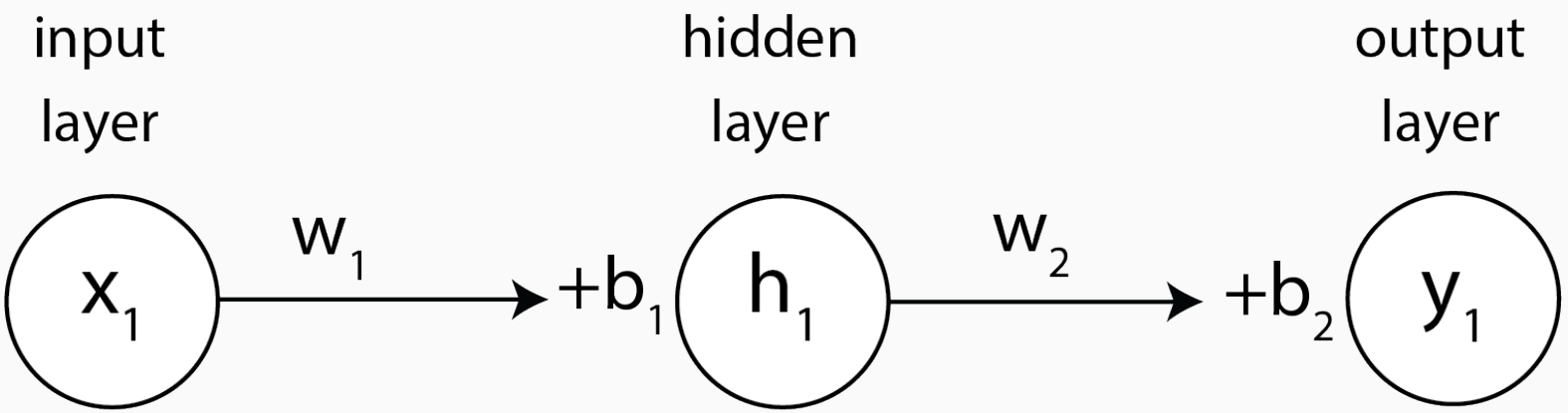
Graphical Neural Networks

- 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 exited about how GNNs might be used in phylogenetics.

Outline

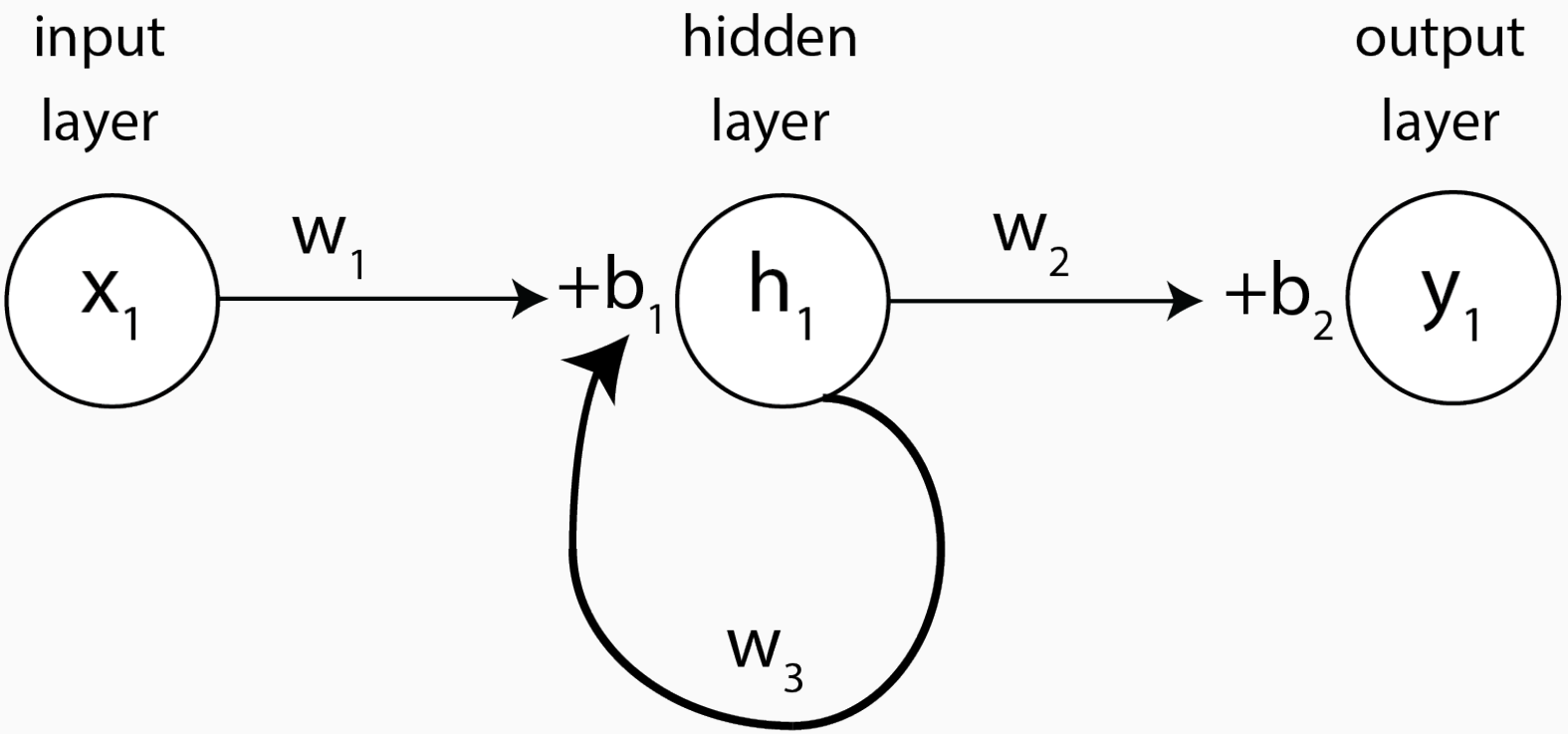
What is a RNN?

My dog, Dexter, loves to chase **squirrels**.

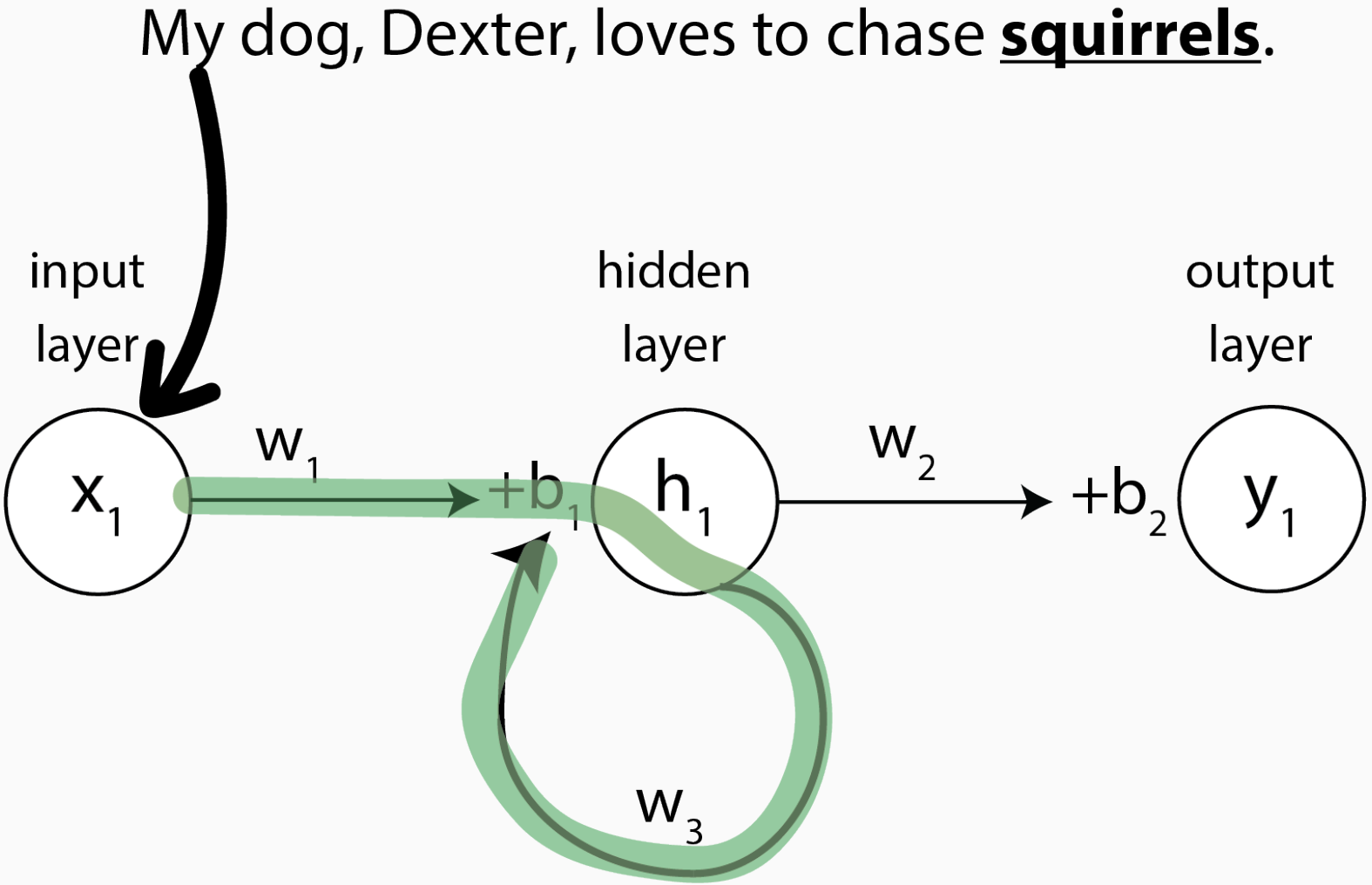


What is a RNN?

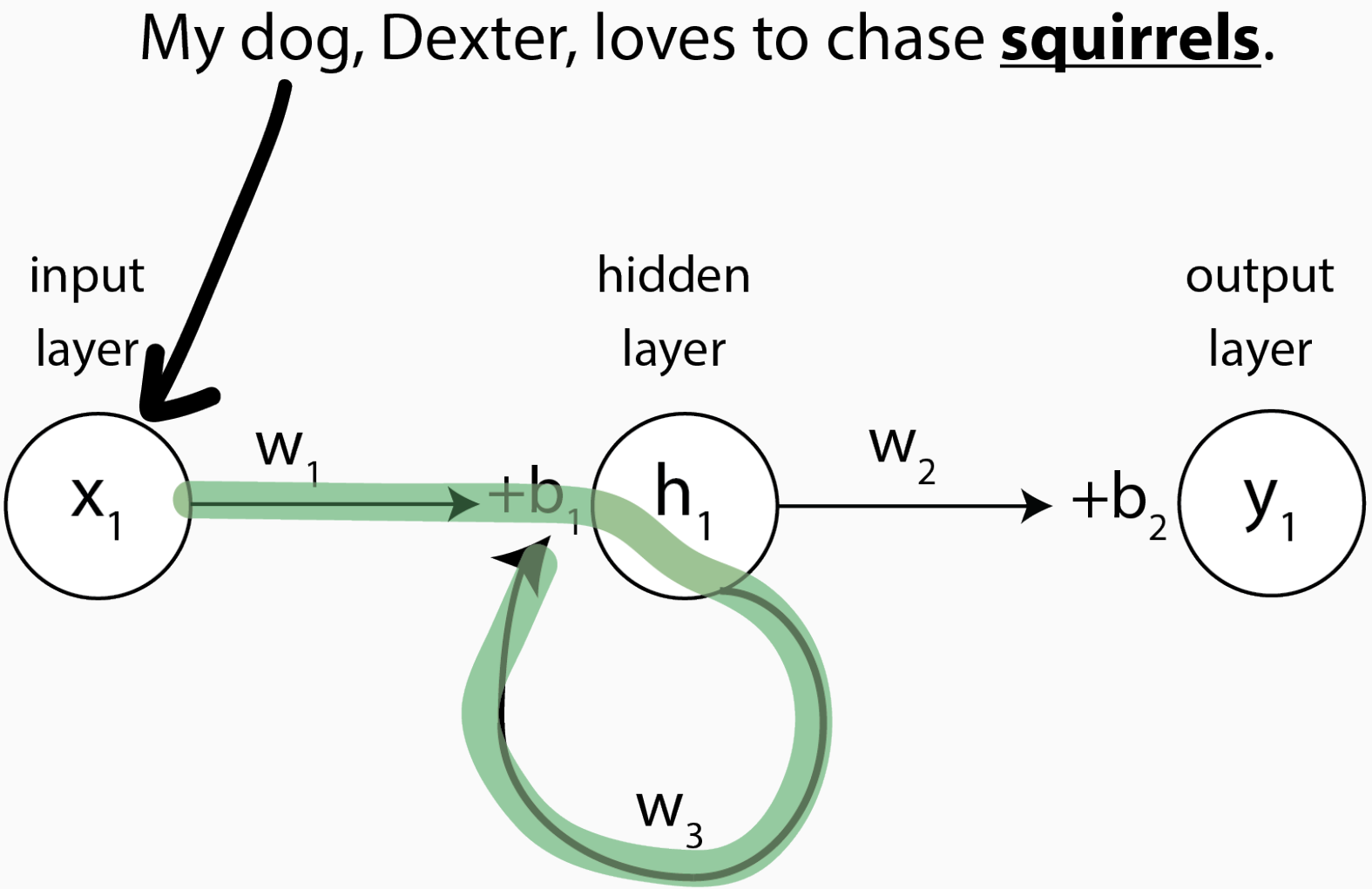
My dog, Dexter, loves to chase **squirrels**.



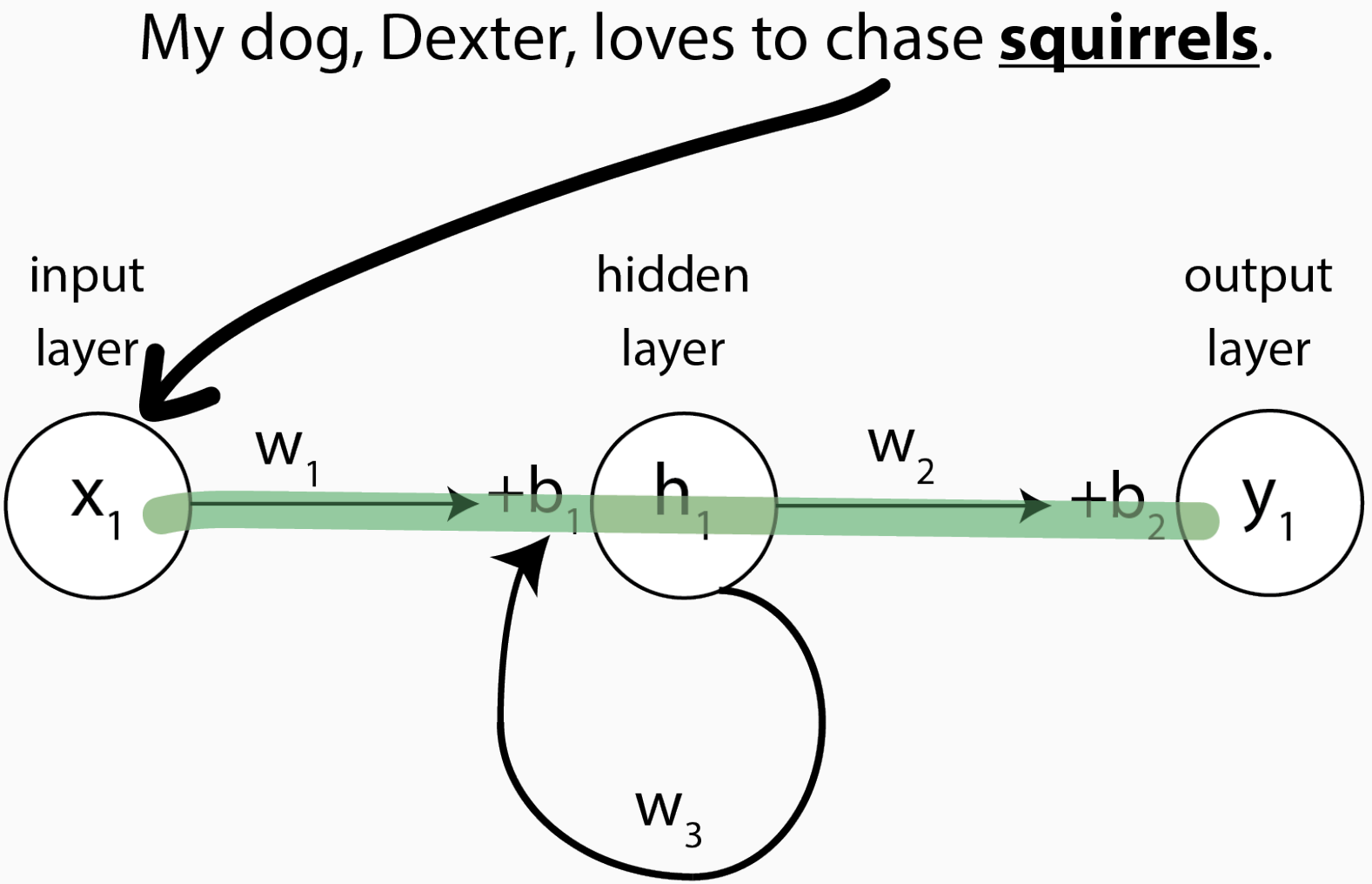
What is a RNN?


My dog, Dexter, loves to chase **squirrels**.
input layer
hidden layer
output layer
x_1
w_1
$+b_1$
h_1
w_2
$+b_2$
y_1
w_3

What is a RNN?


My dog, Dexter, loves to chase **squirrels**.
input layer
hidden layer
output layer
x_1
w_1
$+b_1$
h_1
w_2
$+b_2$
y_1
w_3

What is a RNN?


My dog, Dexter, loves to chase **squirrels**.
input layer
hidden layer
output layer
x_1
w_1
$+b_1$
h_1
w_2
$+b_2$
y_1
w_3

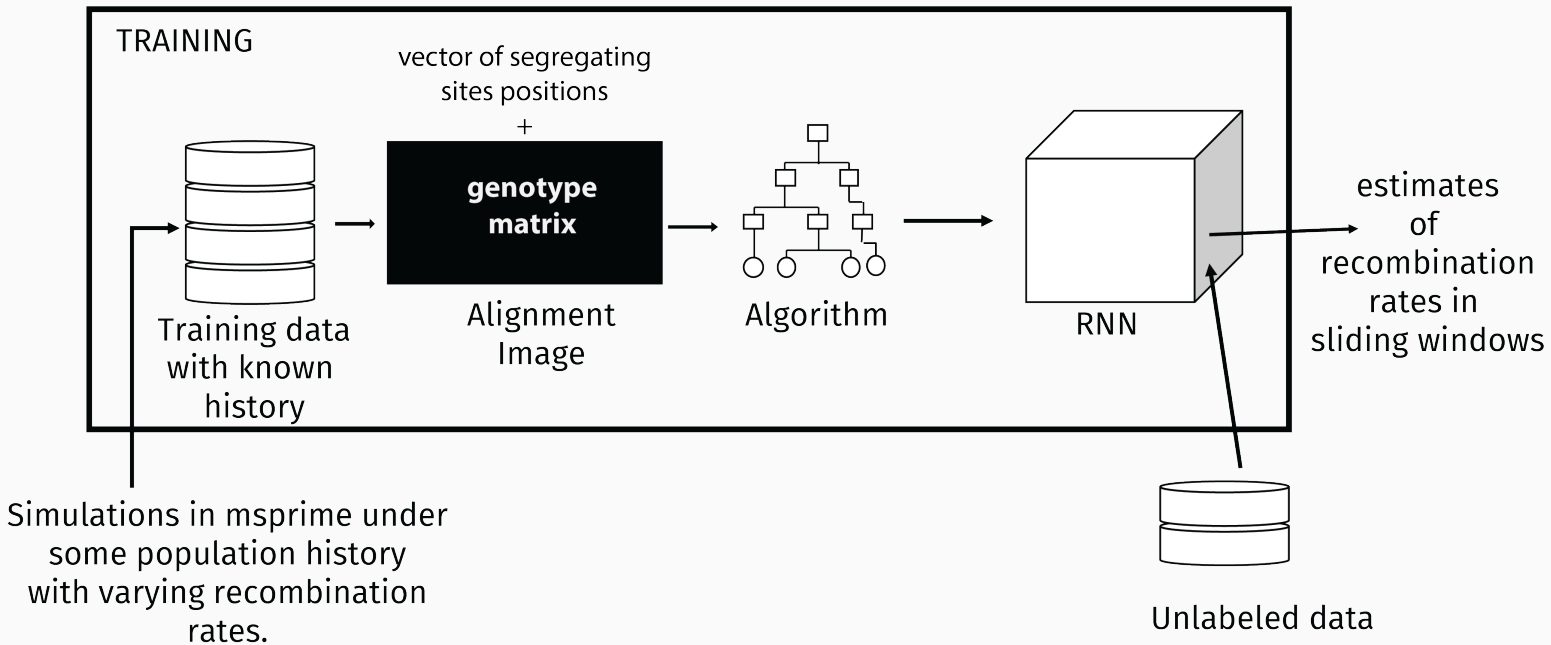
What is a RNN?

- 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.
- Adrion 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., to estimate recombination rates for genomic windows.

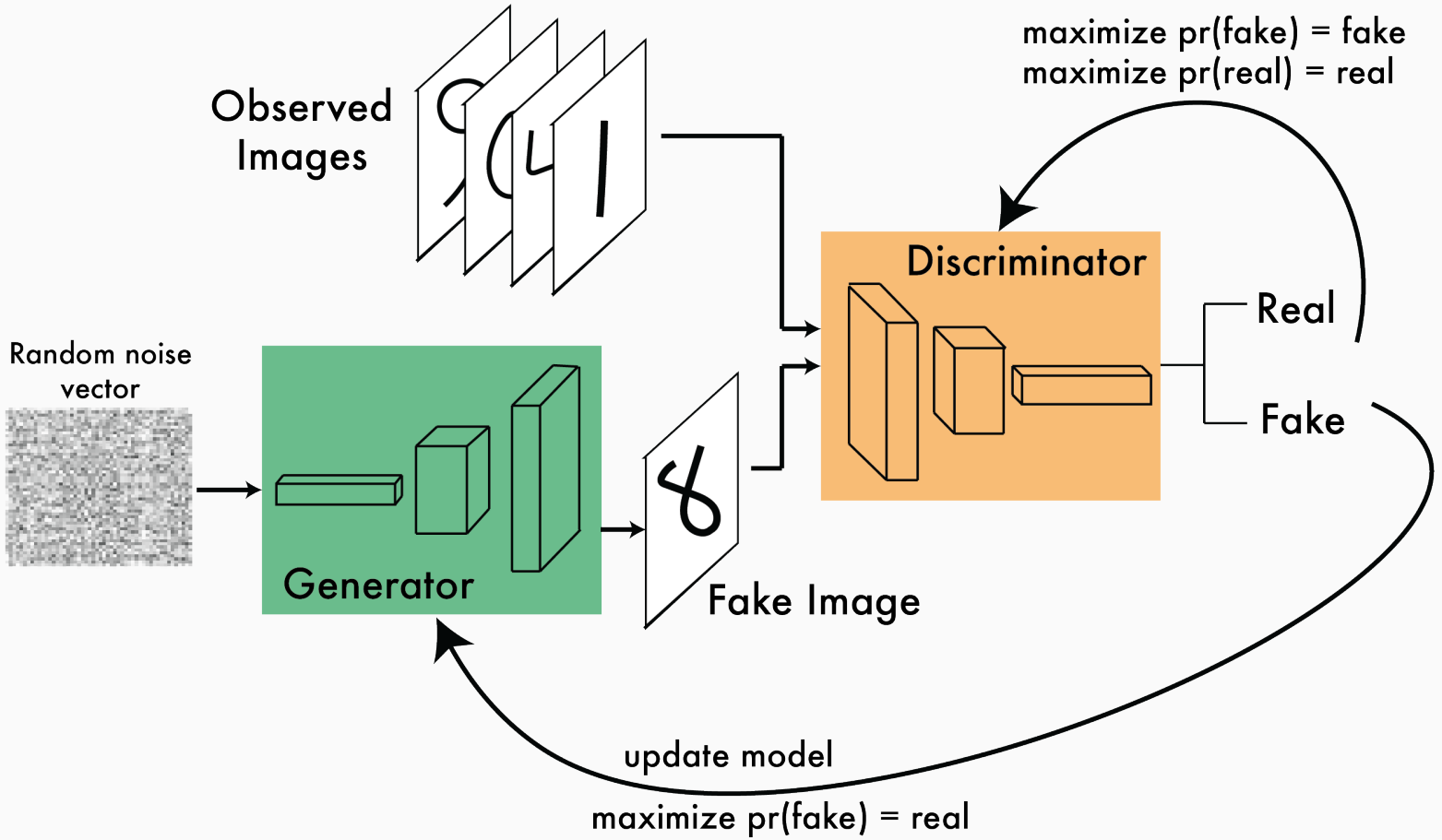
Why might we use a RNN in population genetics?



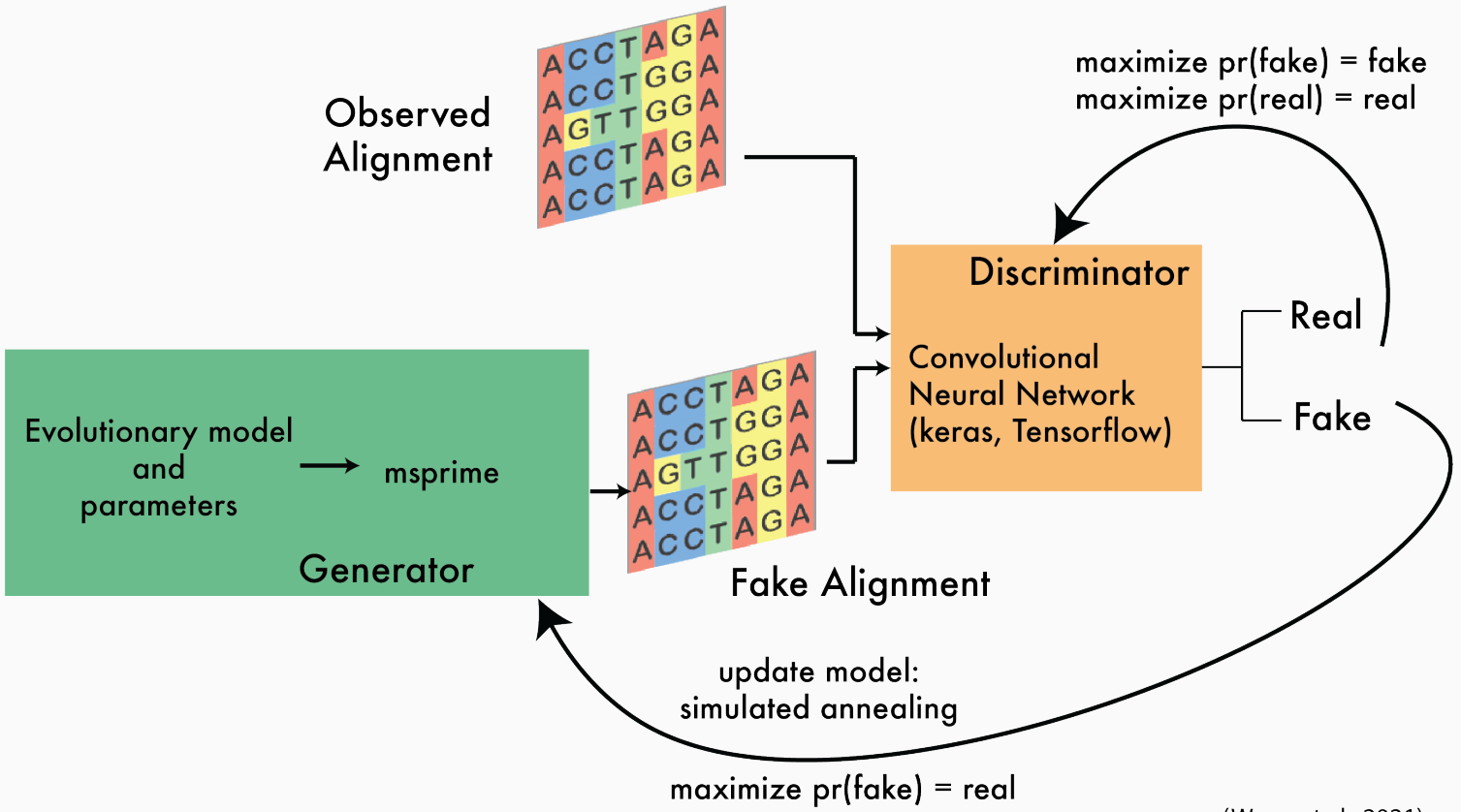
(Adrion et al., 2020)

Outline

What is a Generative Adversarial Network?



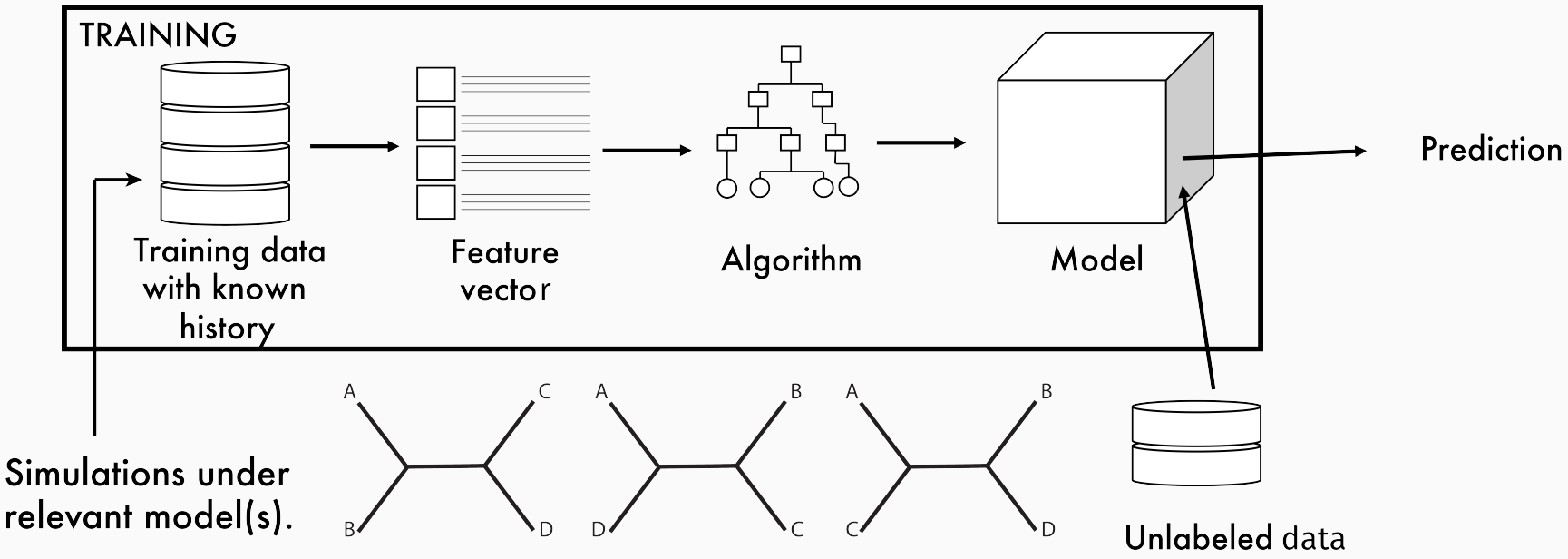
How have GANs been used in population genetics?



(Wang et al., 2021)

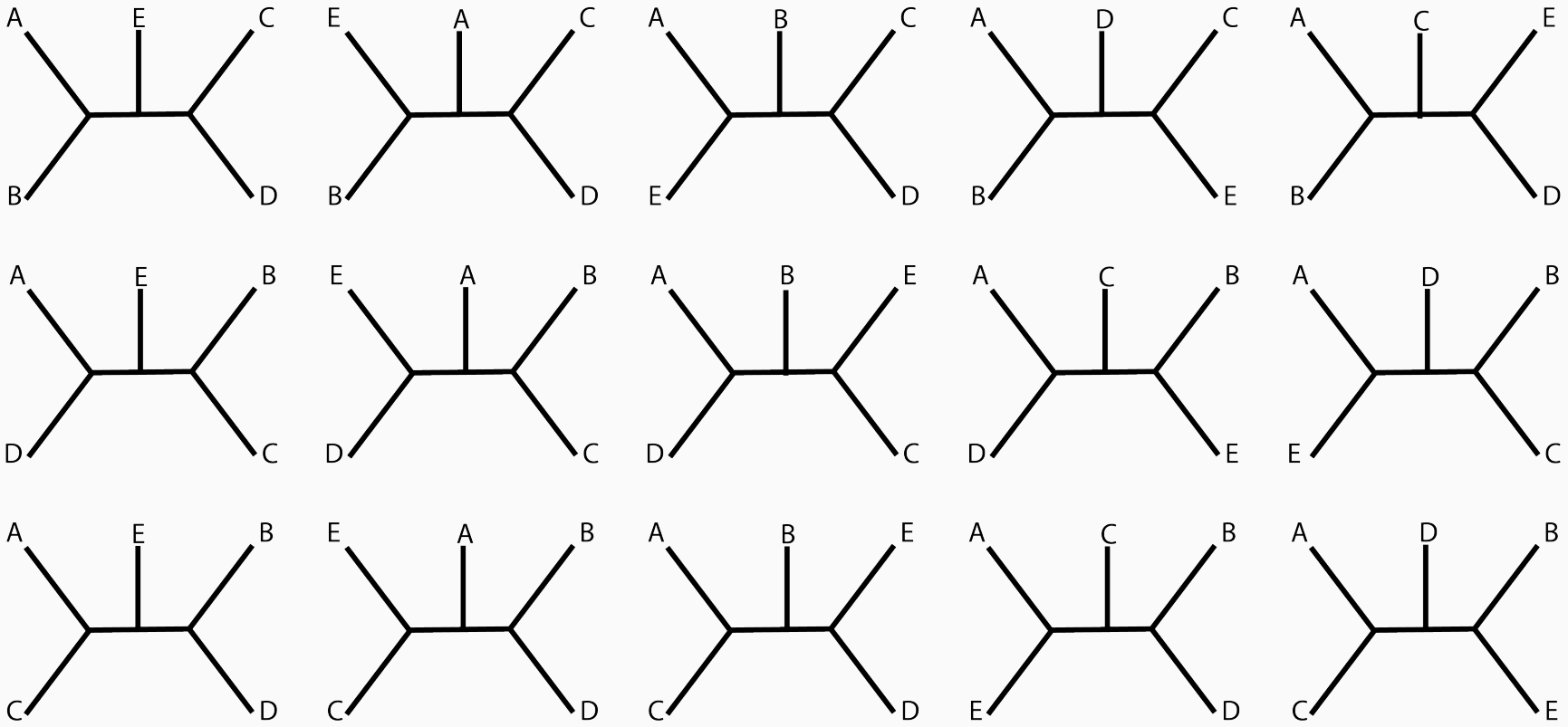
Why might we need a GAN for phylogenetics?

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.



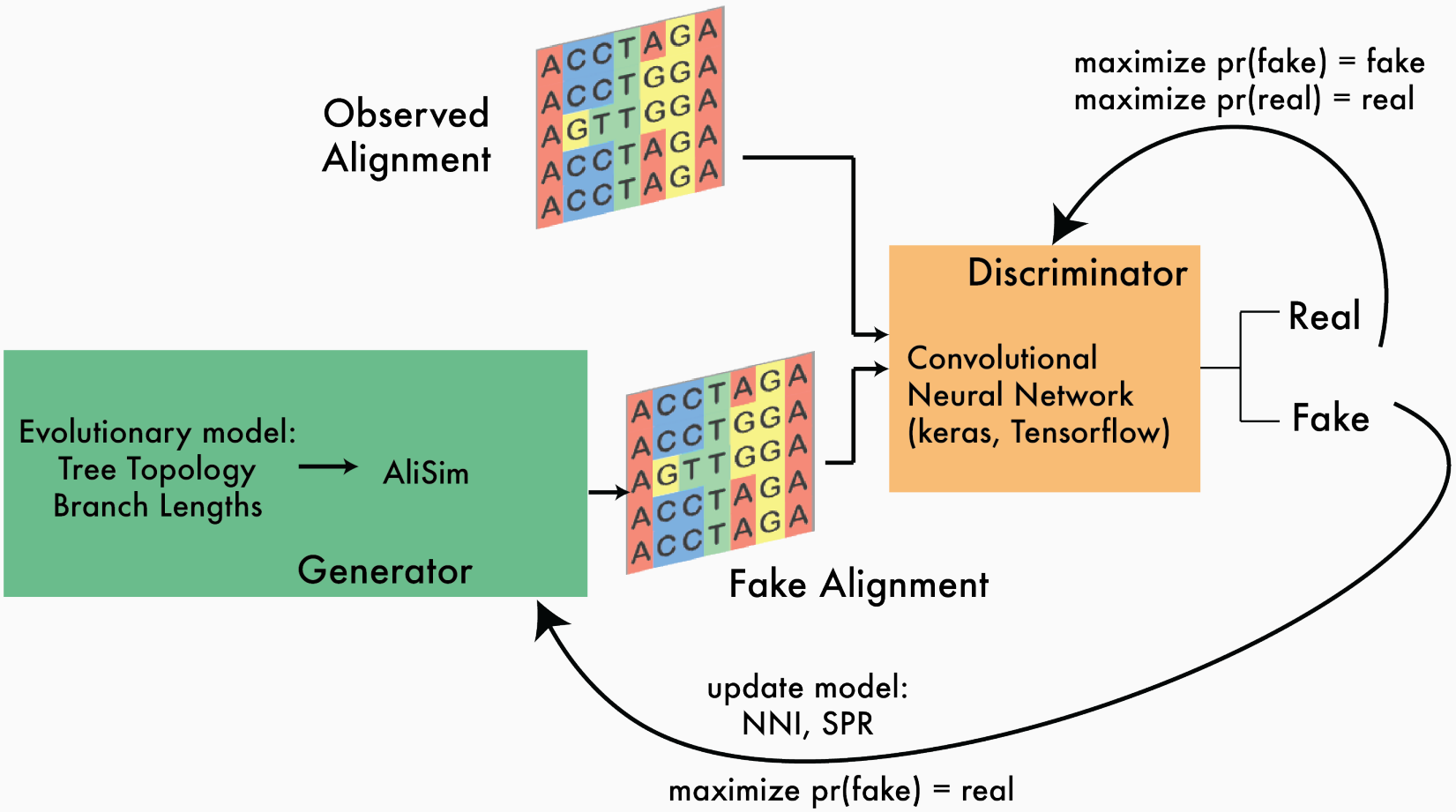
Why might we need a GAN for phylogenetics?

In phylogenetics, the model space is (at a mininum) the tree space.

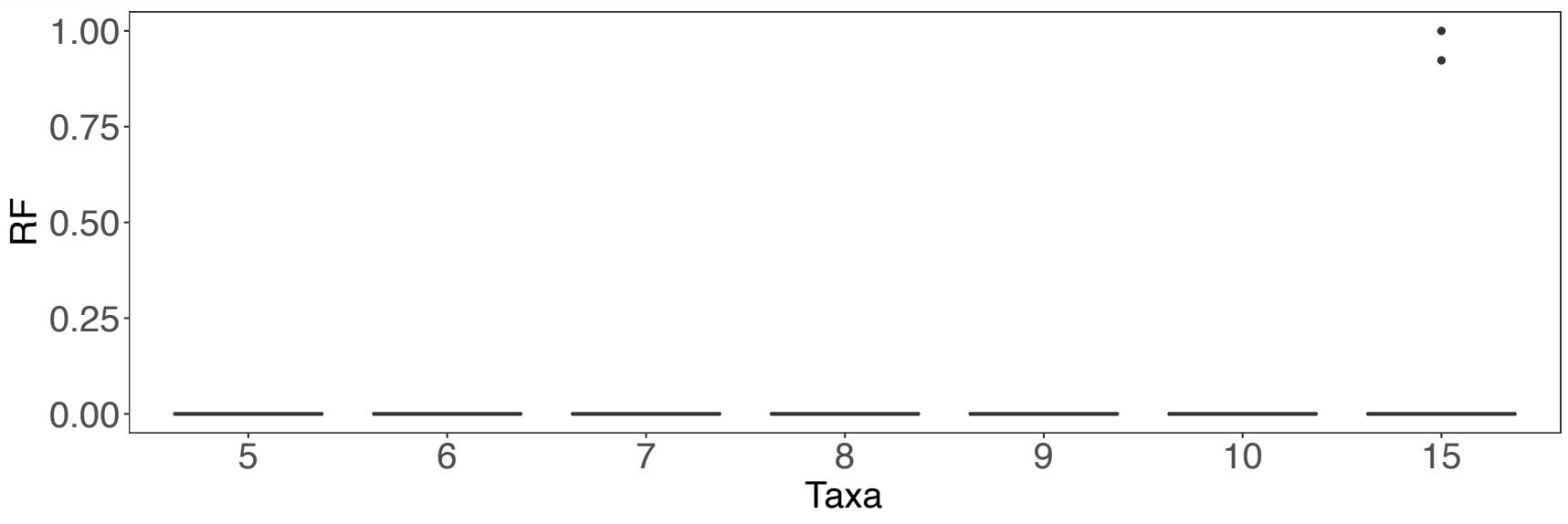
The number of possible trees becomes prohibitively large even with moderate numbers of taxa.

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

phyloGAN


Observed Alignment
maximize pr(fake) = fake
maximize pr(real) = real
Discriminator
Convolutional Neural Network (keras, Tensorflow)
Real
Fake
Evolutionary model:
Tree Topology
Branch Lengths
AliSim
Generator
Fake Alignment
update model:
NNI, SPR
maximize pr(fake) = real

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

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 (2017) and ml4ils.

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 stride, and the type of pooling to use.
- CNNs can directly process images of alignment, bypassing the need to calculate summary statistics.
- We discussed several examples from Flagel et al., (2018), and an approach to infer quartet trees (Suvorov et al., 2020).

Graphical Neural Networks

- 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.
- FCNNs can implement convolutional layers or not.
- Examples include ReLERNN (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 distingiuish generated from real data.
- By using an evolutionary model and parameters as the generator, we can make evolutionary inferences.
- Hyperparameters include those of the discriminator (often a neural network), along with the hyperparameters controlling proposals of the generator.
- Examples include pg-GAN (Wang et al., 2021) and phyloGAN (Smith and Hahn, 2022).

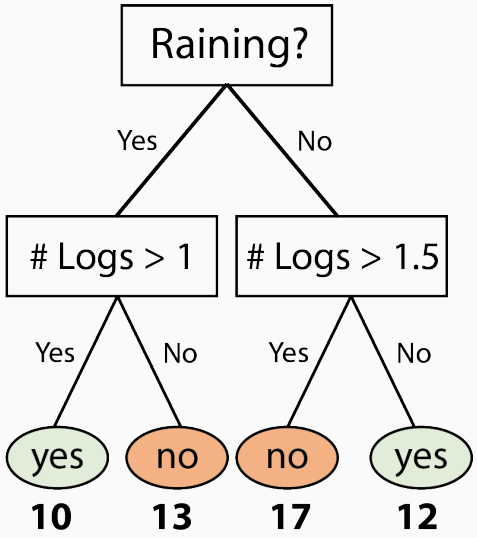
Challenges and Future Directions

Outline

How do I avoid overfitting?

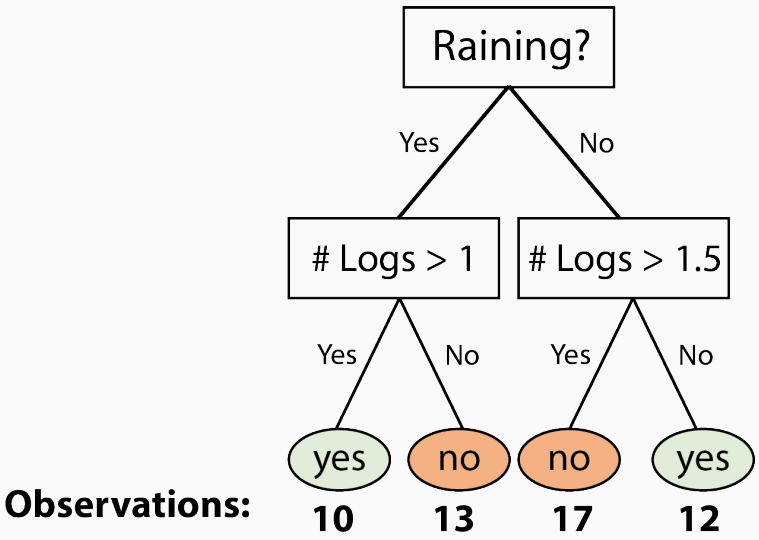
- **Overfitting** occurs when the model gives accurate predictions for the training data, but not new data.
- We can attempt to avoid overfitting using several approaches:

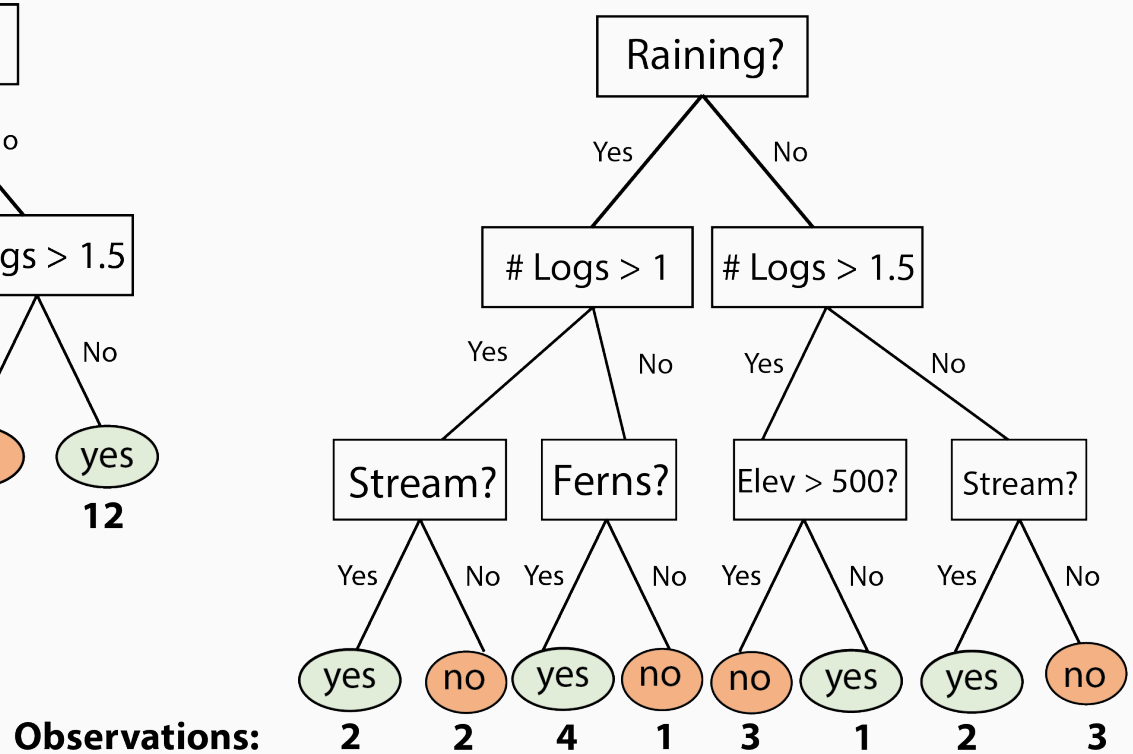
Overfitting: Random Forests



Observations:

Overfitting: Random Forests

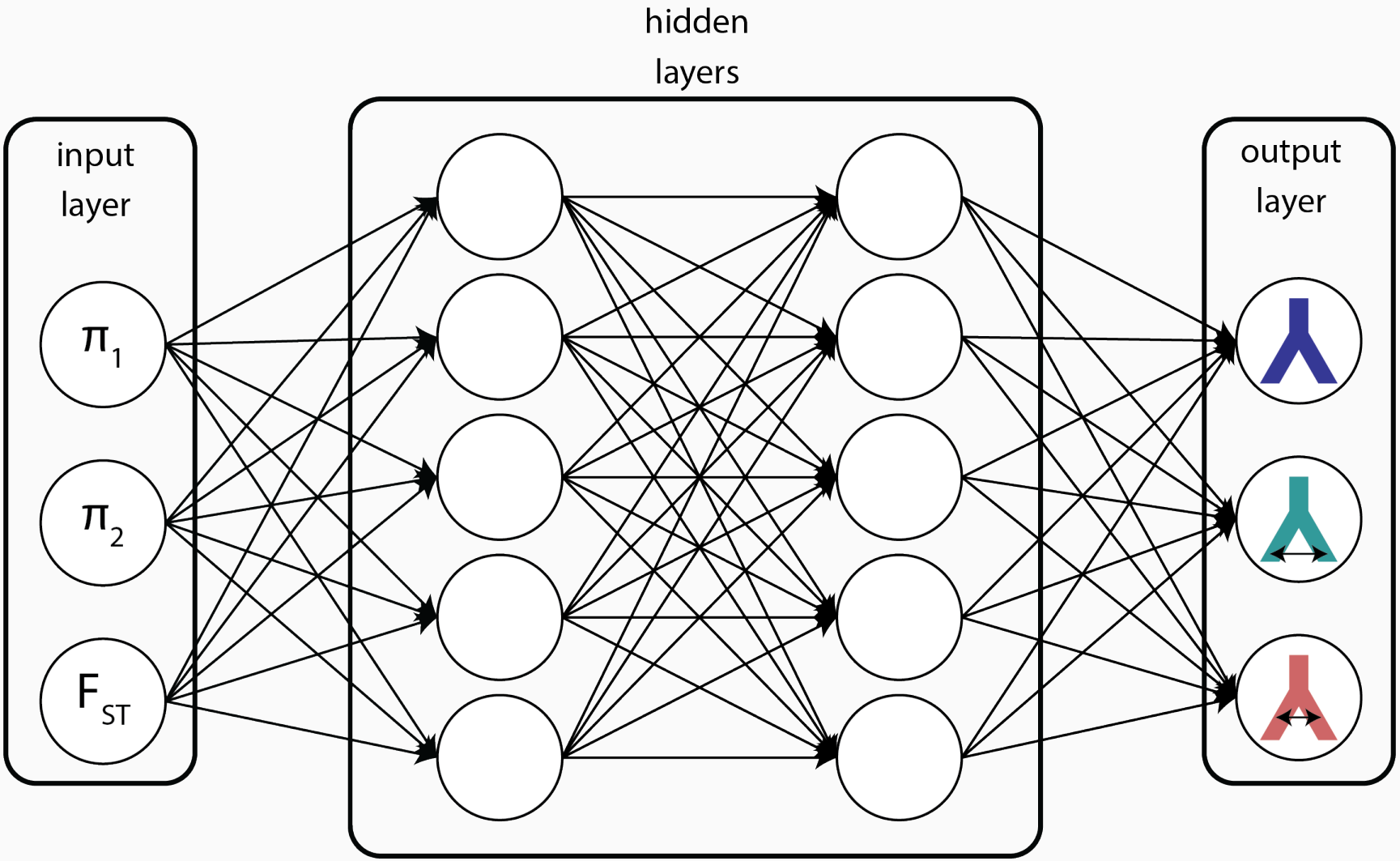




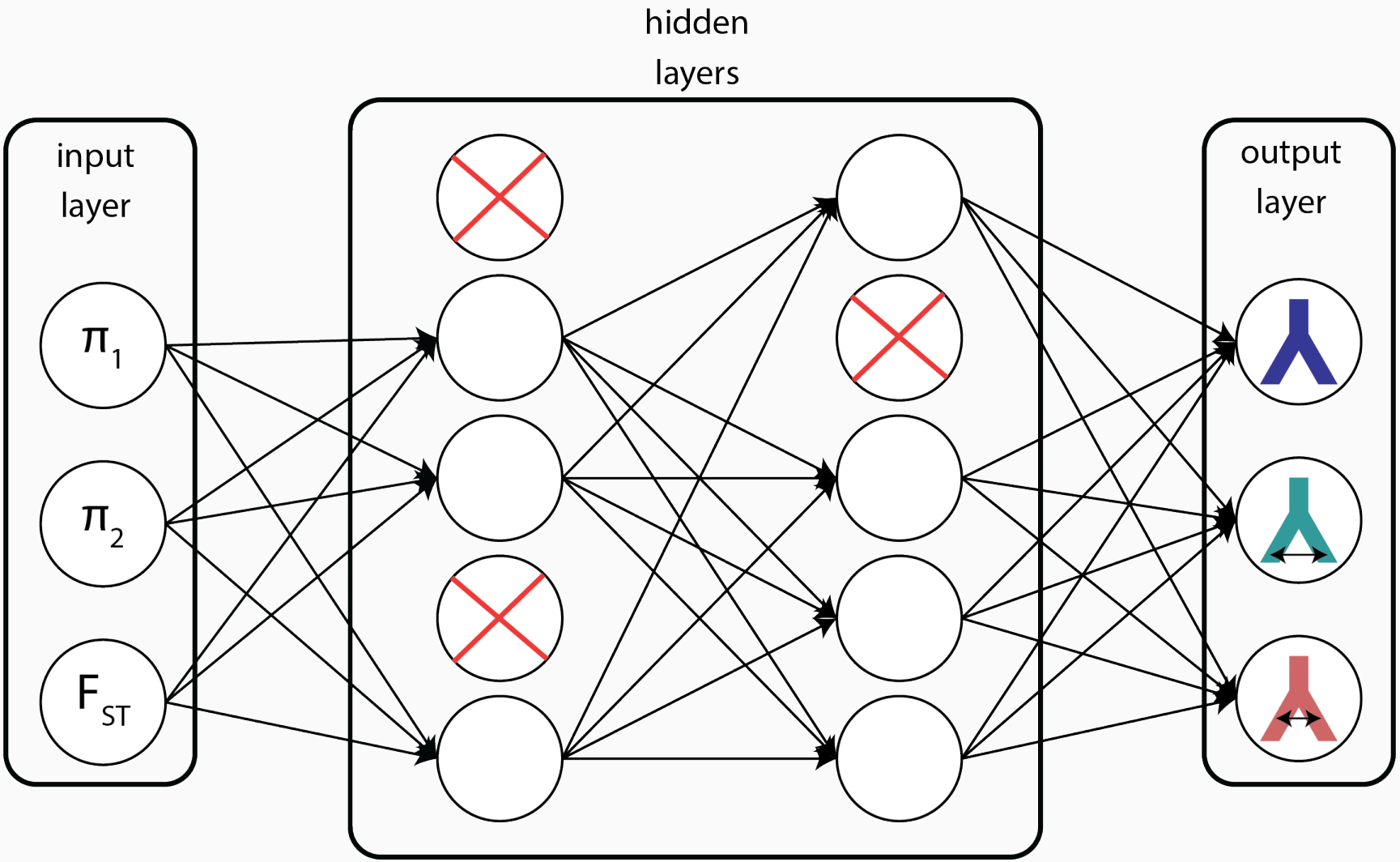
How do I avoid overfitting?

- **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

How do I choose my hyperparameters

Hyperparameters can be optimized using several approaches:

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

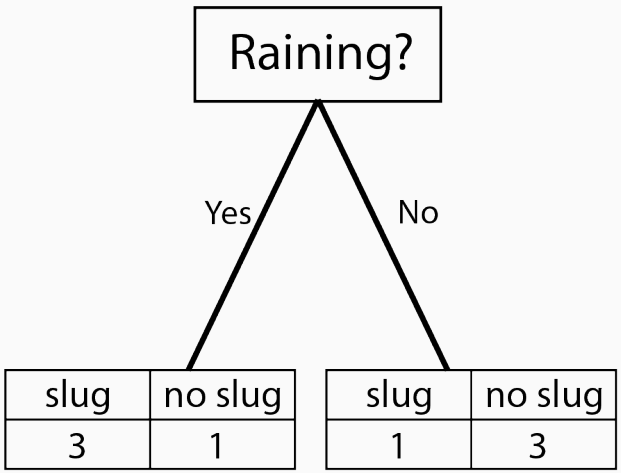
Outline

The Black Box

Can we understand how features are driving predictions?

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

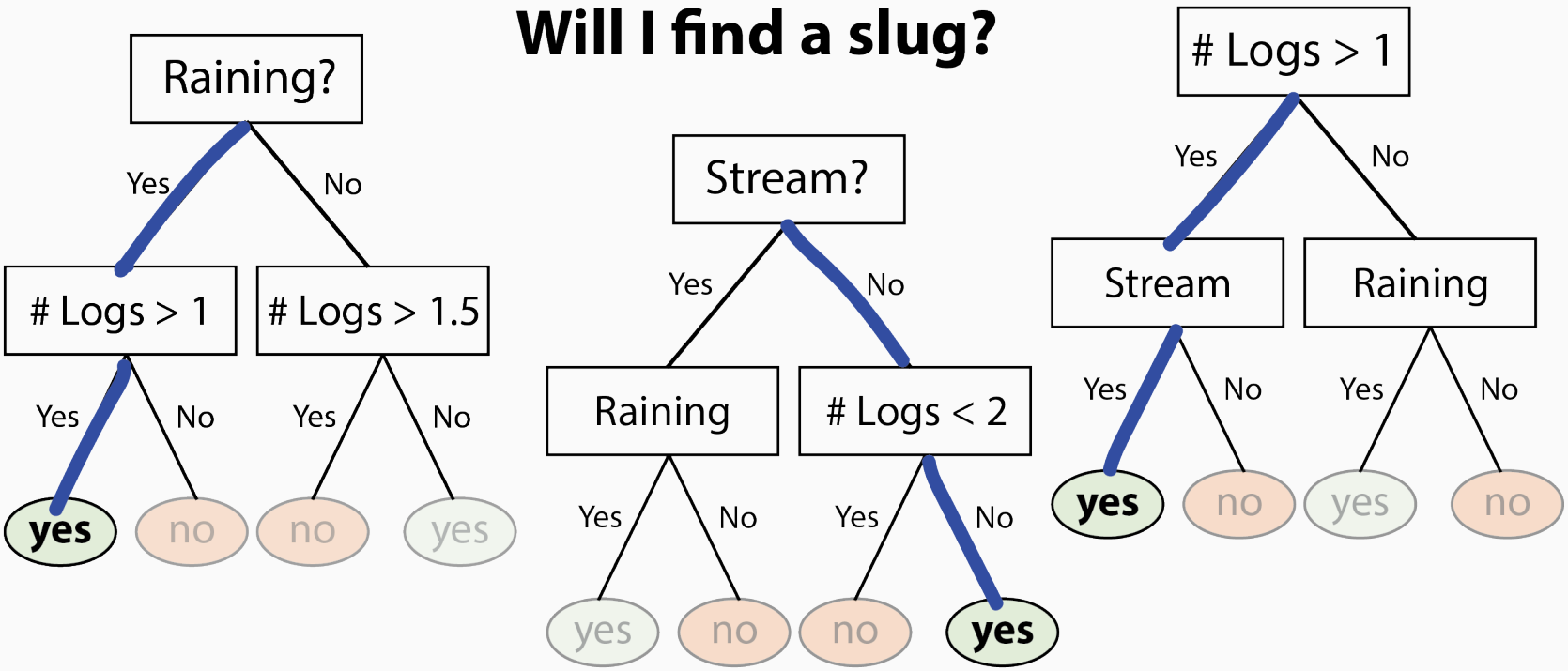
measure of node purity

0: perfectly pure

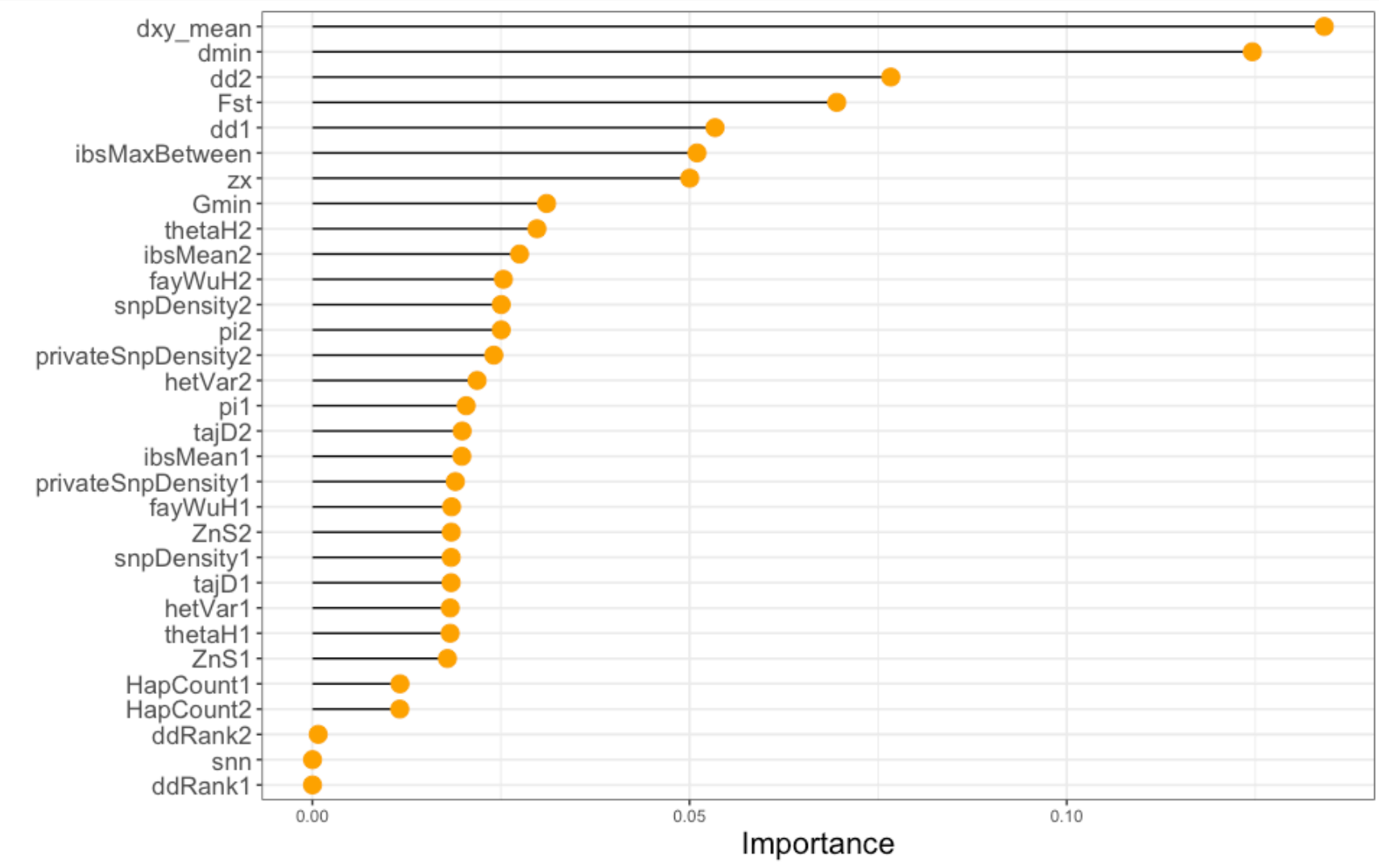
1: elements randomly distributed

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

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

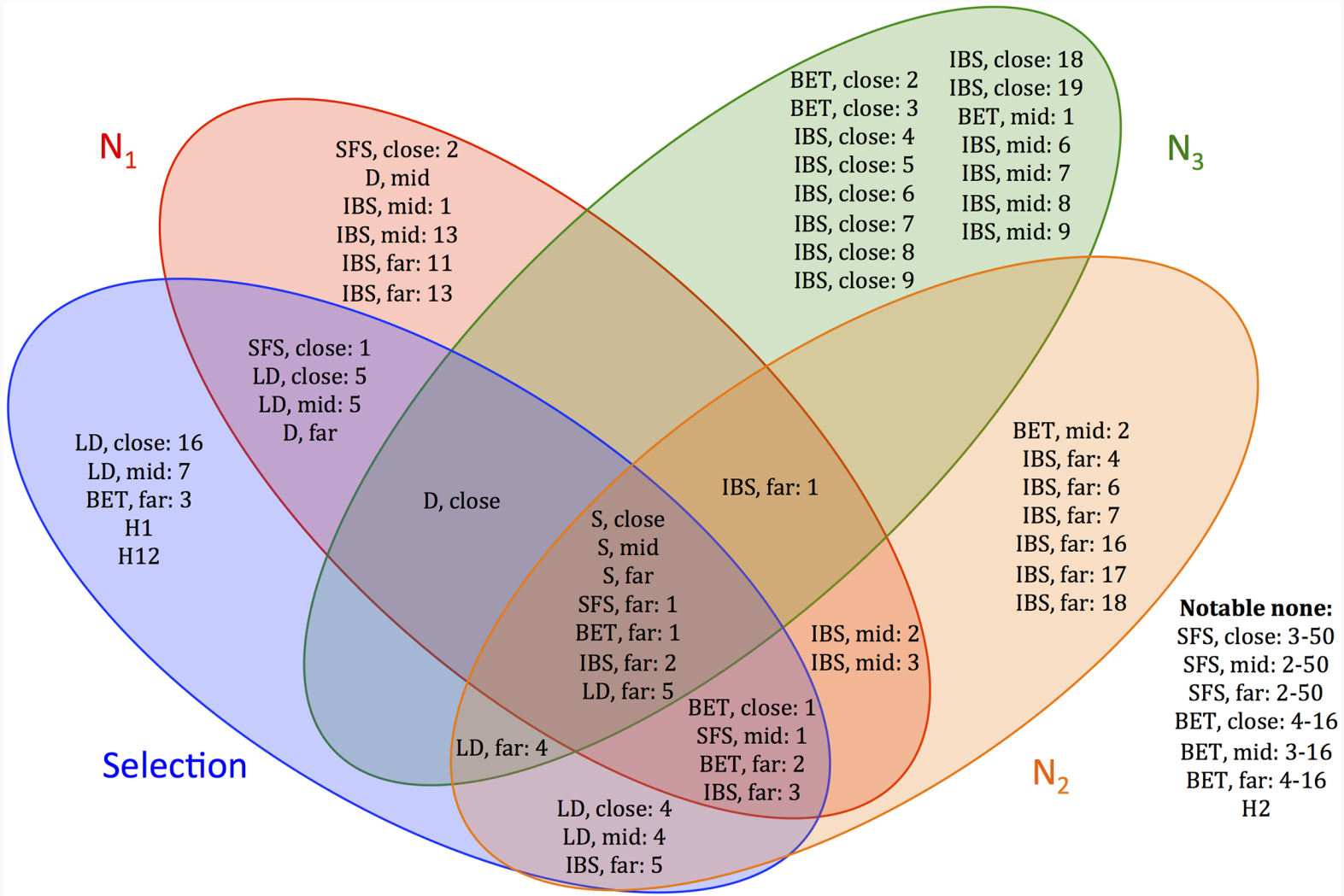


Figure 5 from Sheehan and Song, 2016

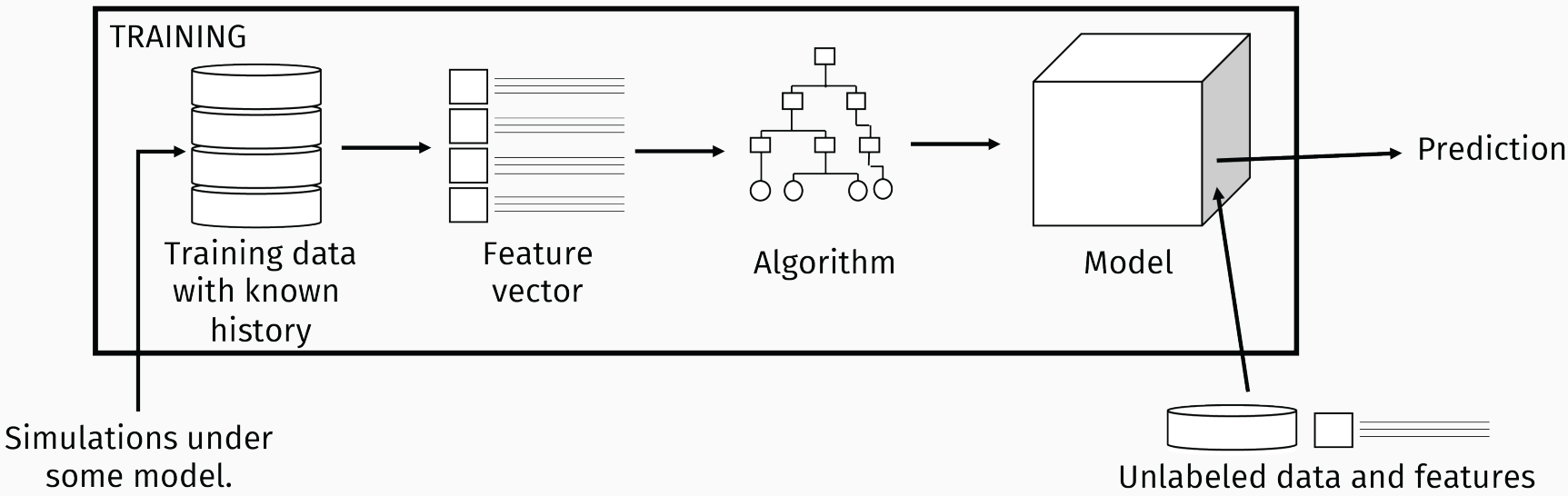
Does this really solve the ”Black Box” problem?

- We should always prefer methods that allow us to estimate a likelihood if we have them! Explicit statistical models are great!

- ***However***, we often cannot do this for complex models. In cases where we are worried that a simple model will be misleading, machine learning can offer a power and flexible alternative.

Outline

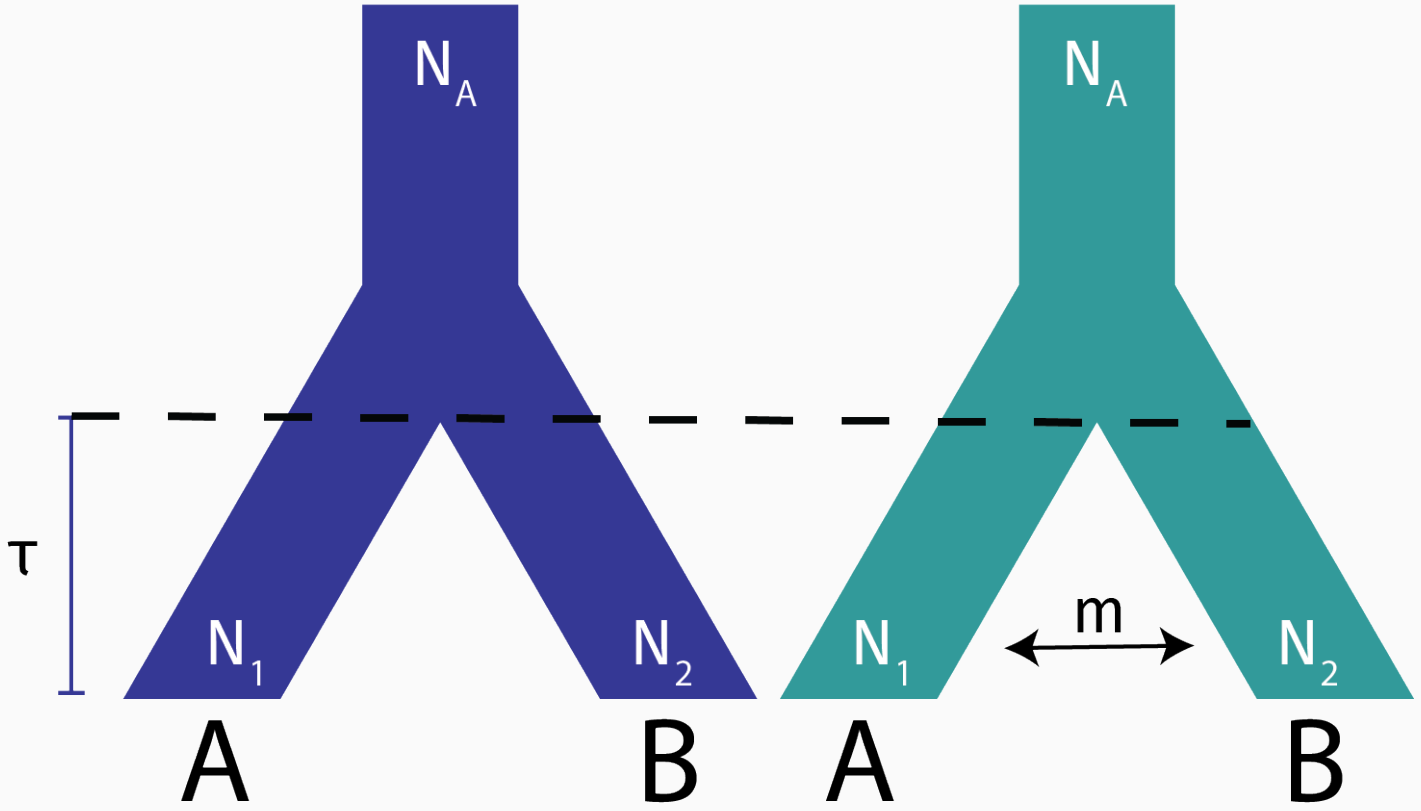
Simulation Misspecification



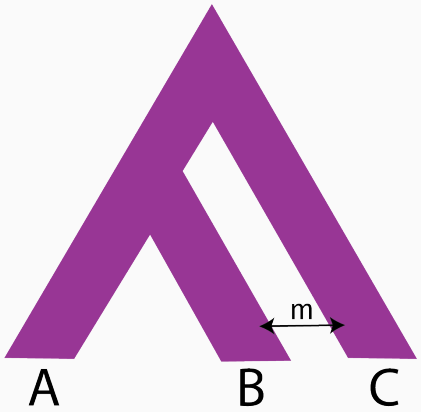
Simulation Misspecification

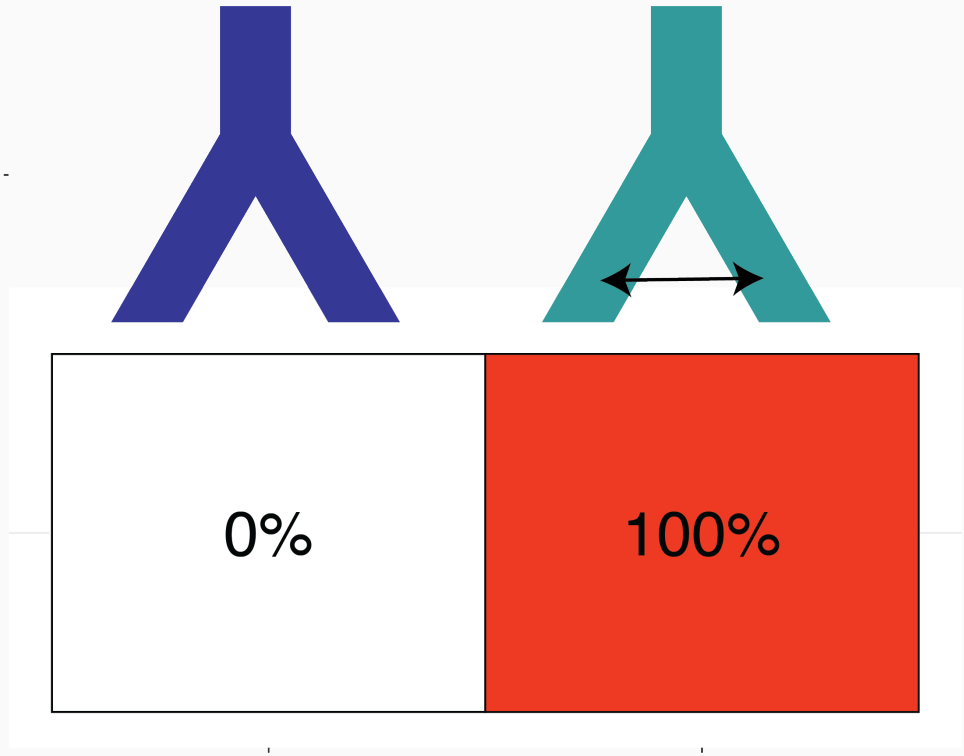
What if our simulations aren't realistic?

Simulations:



Simulation Misspecification



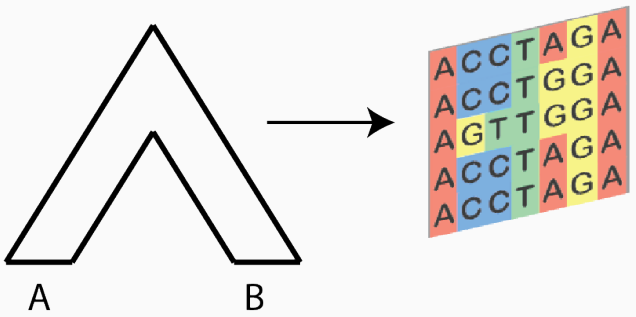
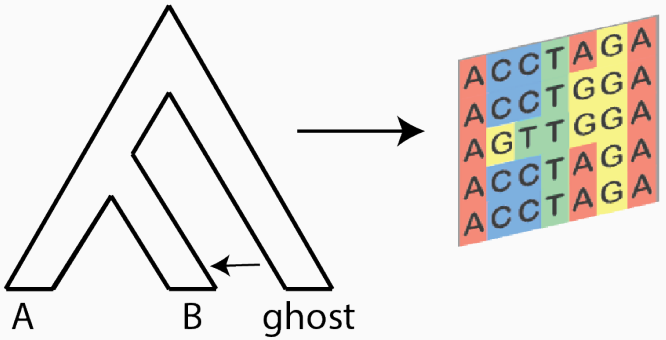


Domain Shifts

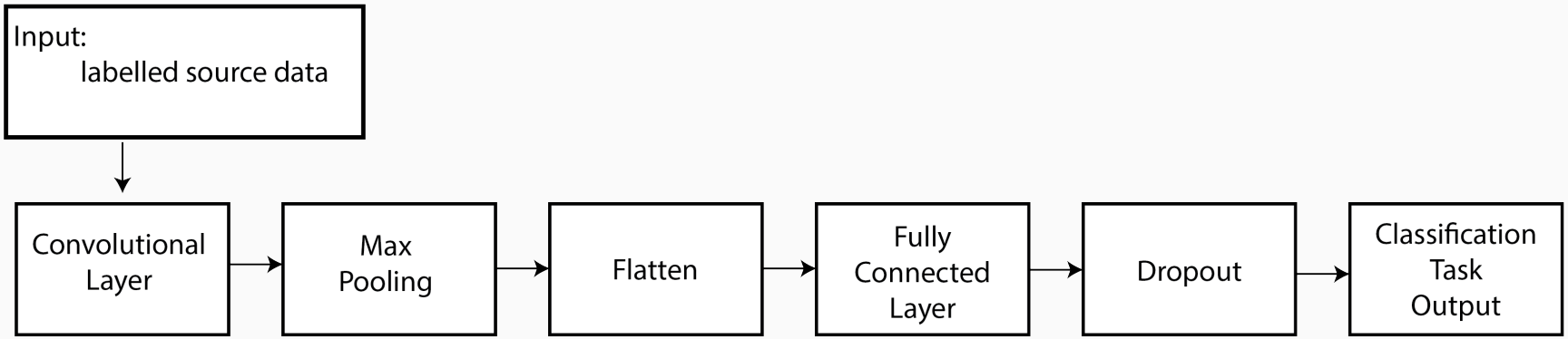
Domain shifts occur when training data and test data arise from different distributions.

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

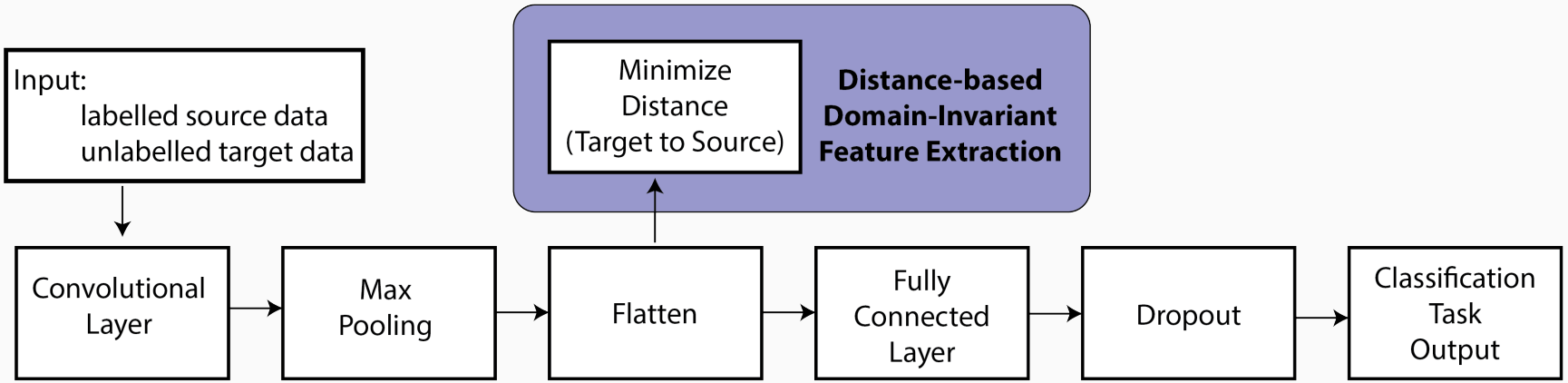
Domain Shifts

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

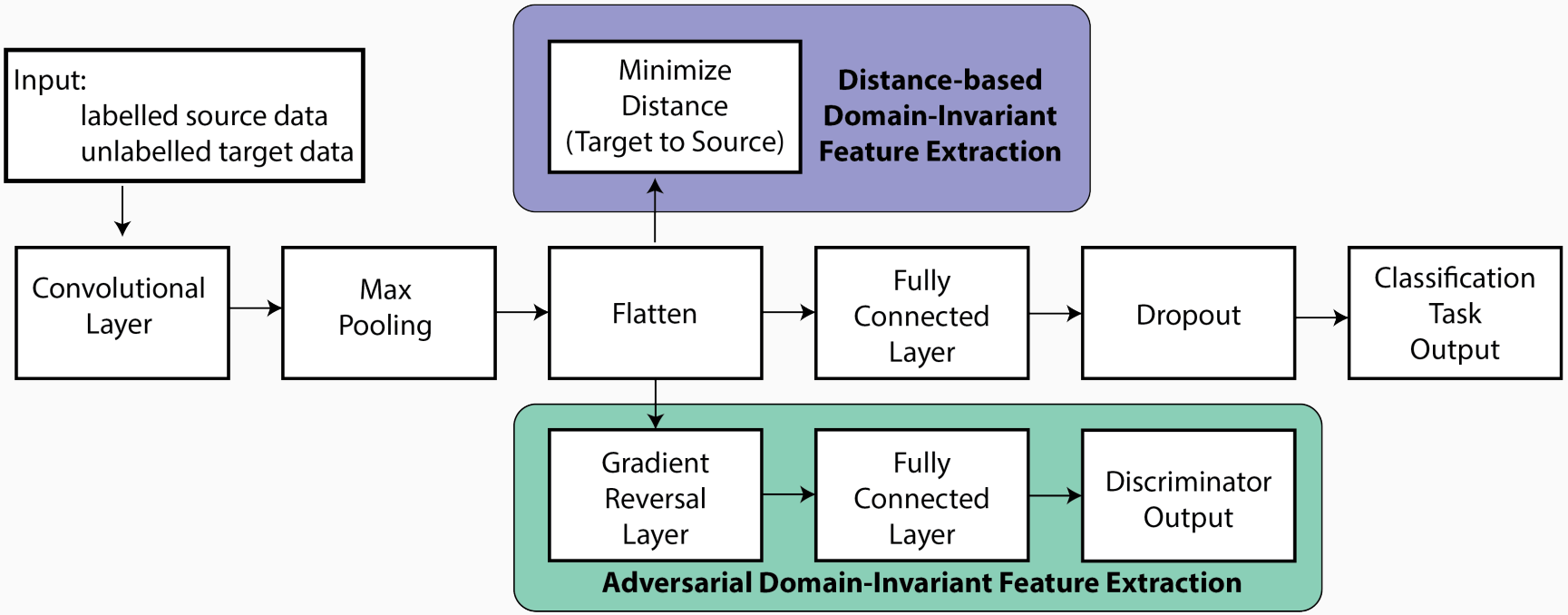
Domain Adaptation



Domain Adaptation



Domain Adaptation

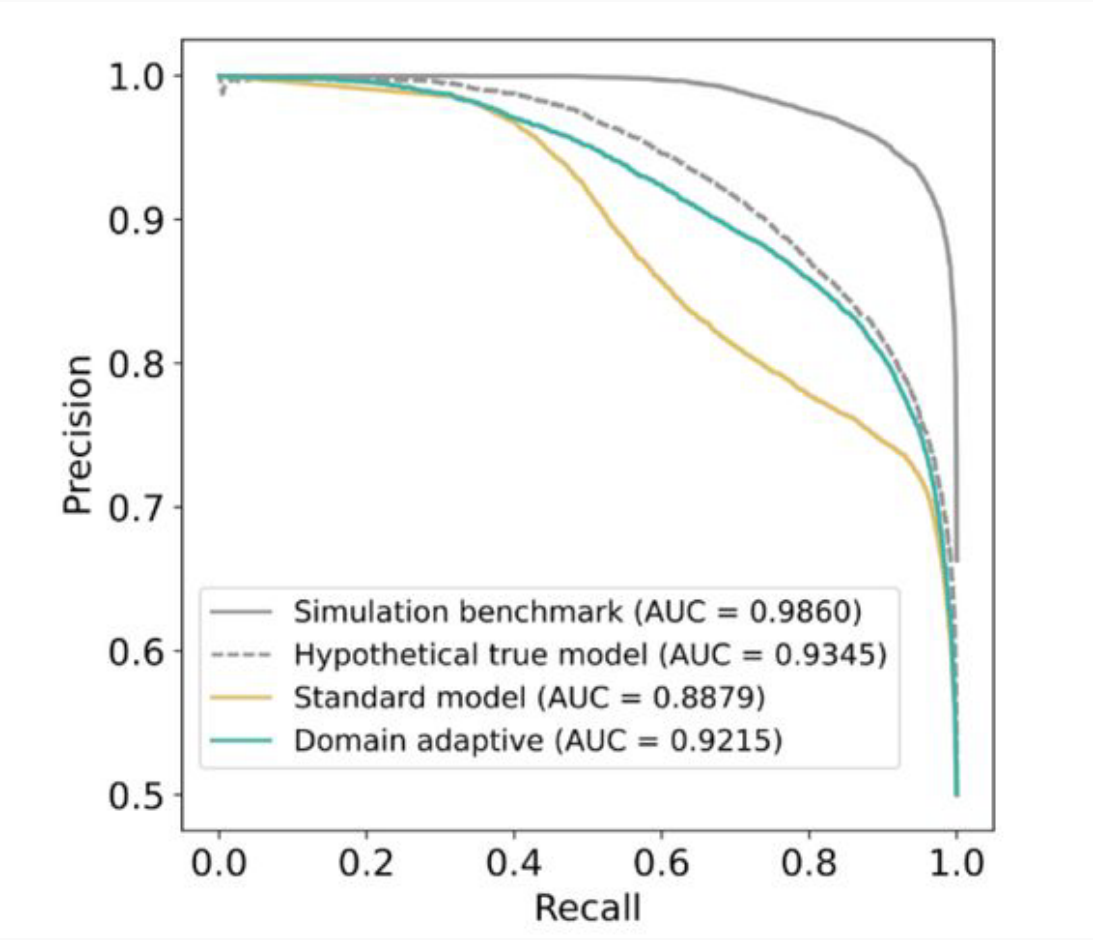


Domain Adaptation

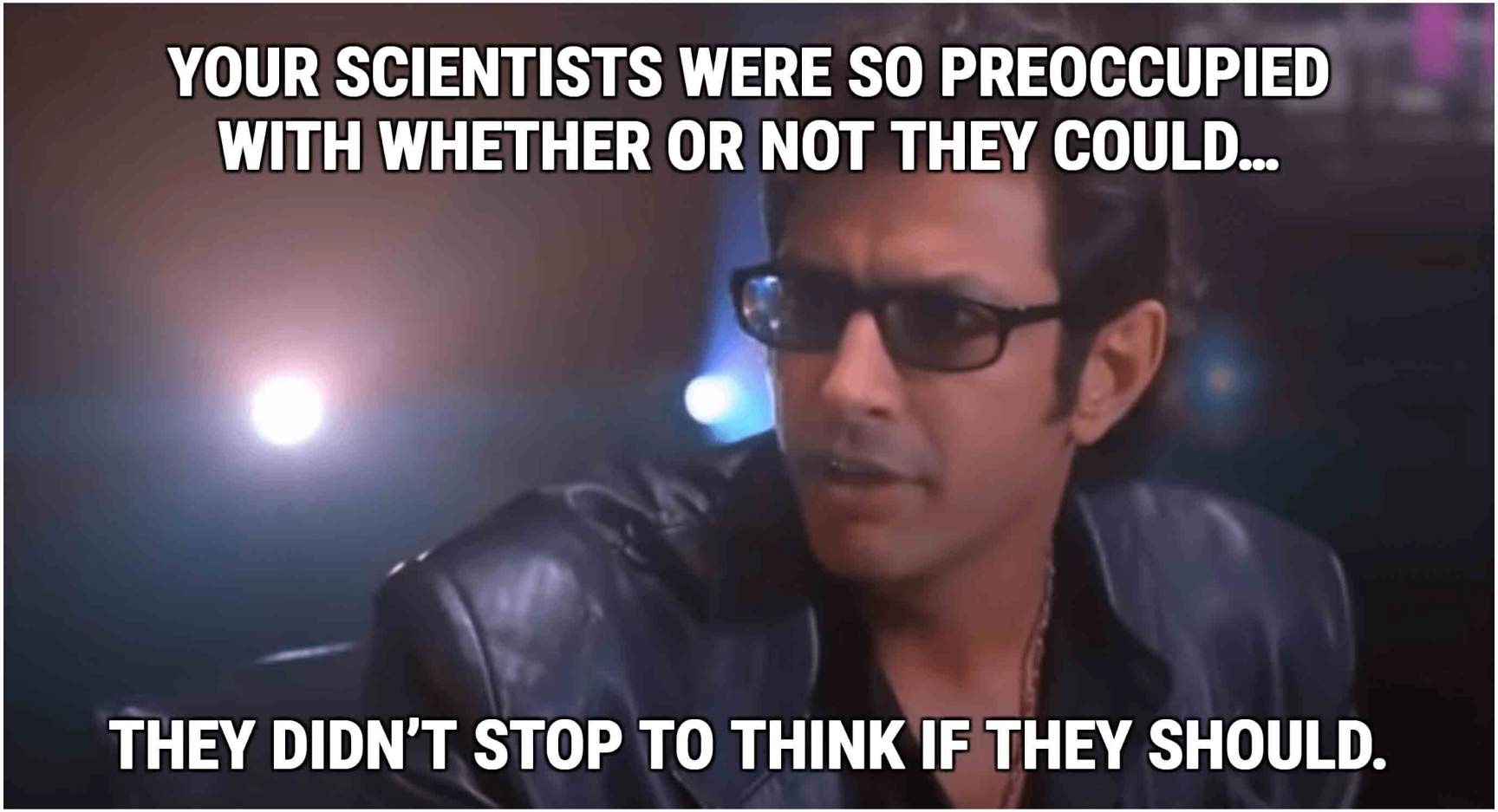
Mo and Siepel (2023) used domain adaptation.

Goal: to estimate the selection coefficient

Misspecification: Background Selection



A cautionary note



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