CS 68: Bioinformatics

Prof. Sara Mathieson

Spring 2018

Swarthmore College

Outline: April 27

- Continue: machine learning for biology
- Convolutional neural networks (CNNs) + applications
- Previous work: Approximate Bayesian Computation (ABC)

Notes:

- I will post readings for our ethics discussions next week (Mon/Wed)
- Attendance and discussion counts toward participation
- Short readings, spend 10-15min (more if you like)

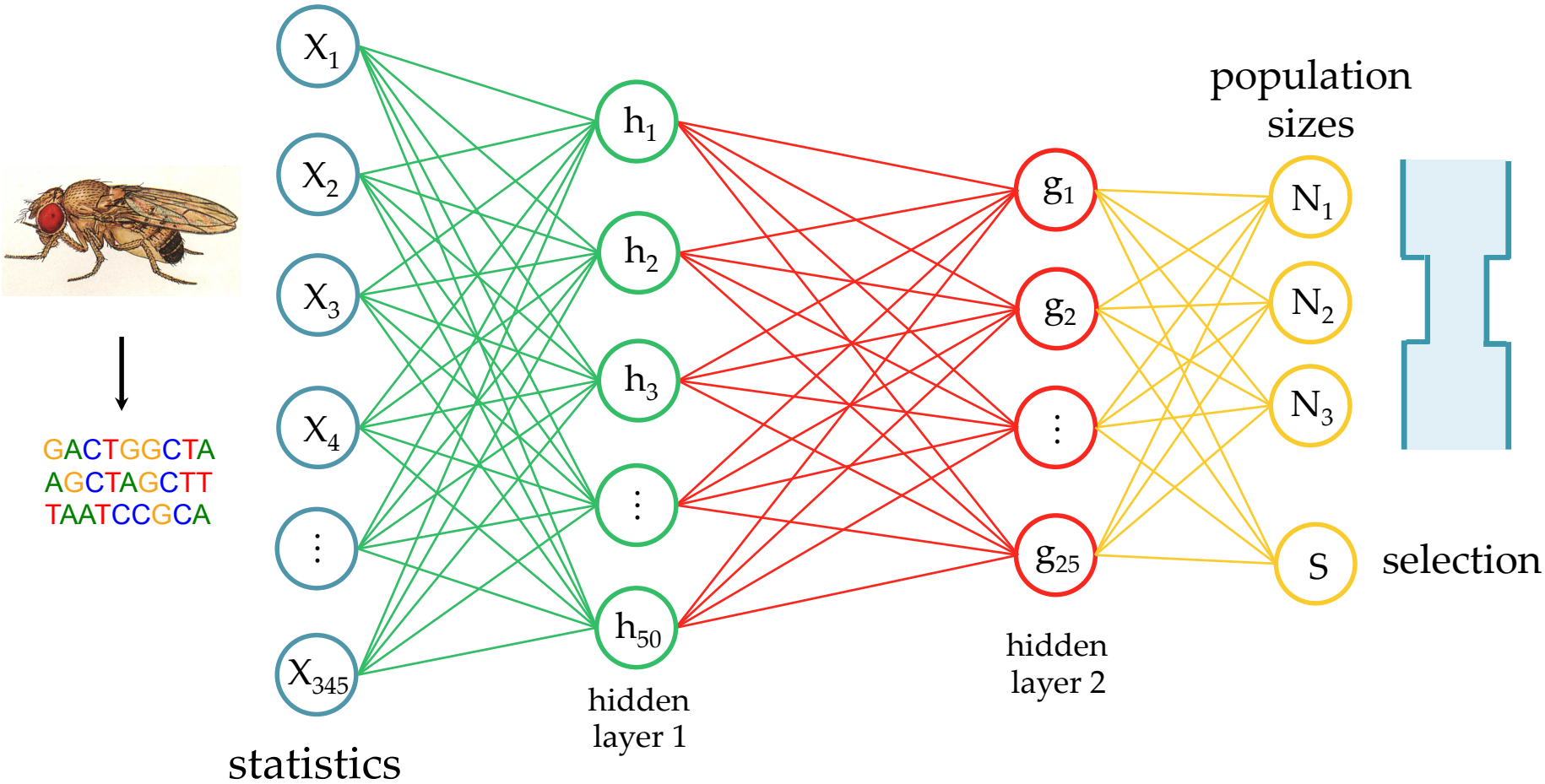
Application of Deep Learning to Population Genetics

Summary statistics as features

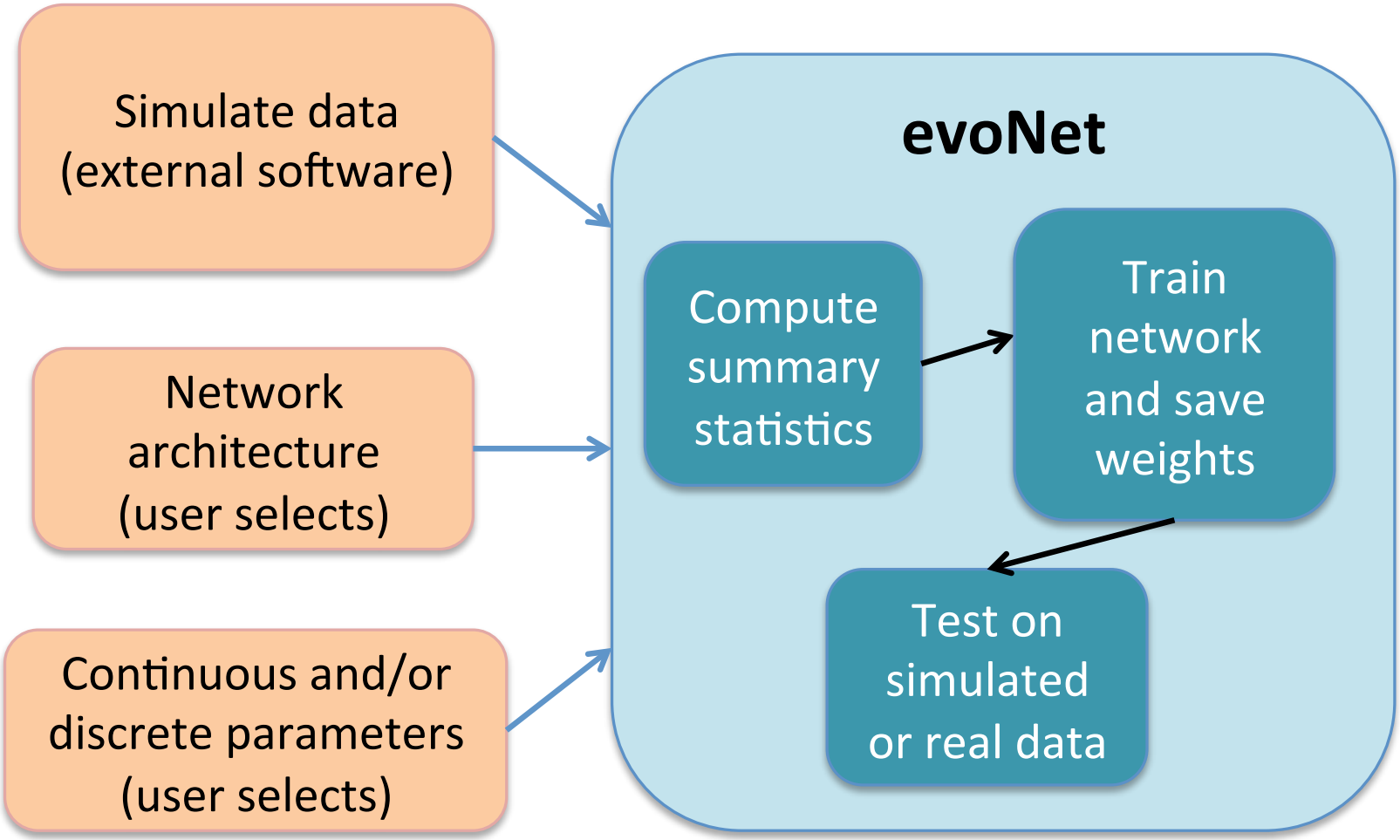
- Number of segregating sites **3 stats**
- Tajima's D **3 stats**
- Folded site frequency spectrum (SFS) **150 stats**
- Length distribution between segregating sites **48 stats**
- Identity-by-state (IBS) tract length distribution **90 stats**
- Linkage disequilibrium (LD) distributions **48 stats**
- Haplotype frequency statistics **3 stats**

= 345 features total

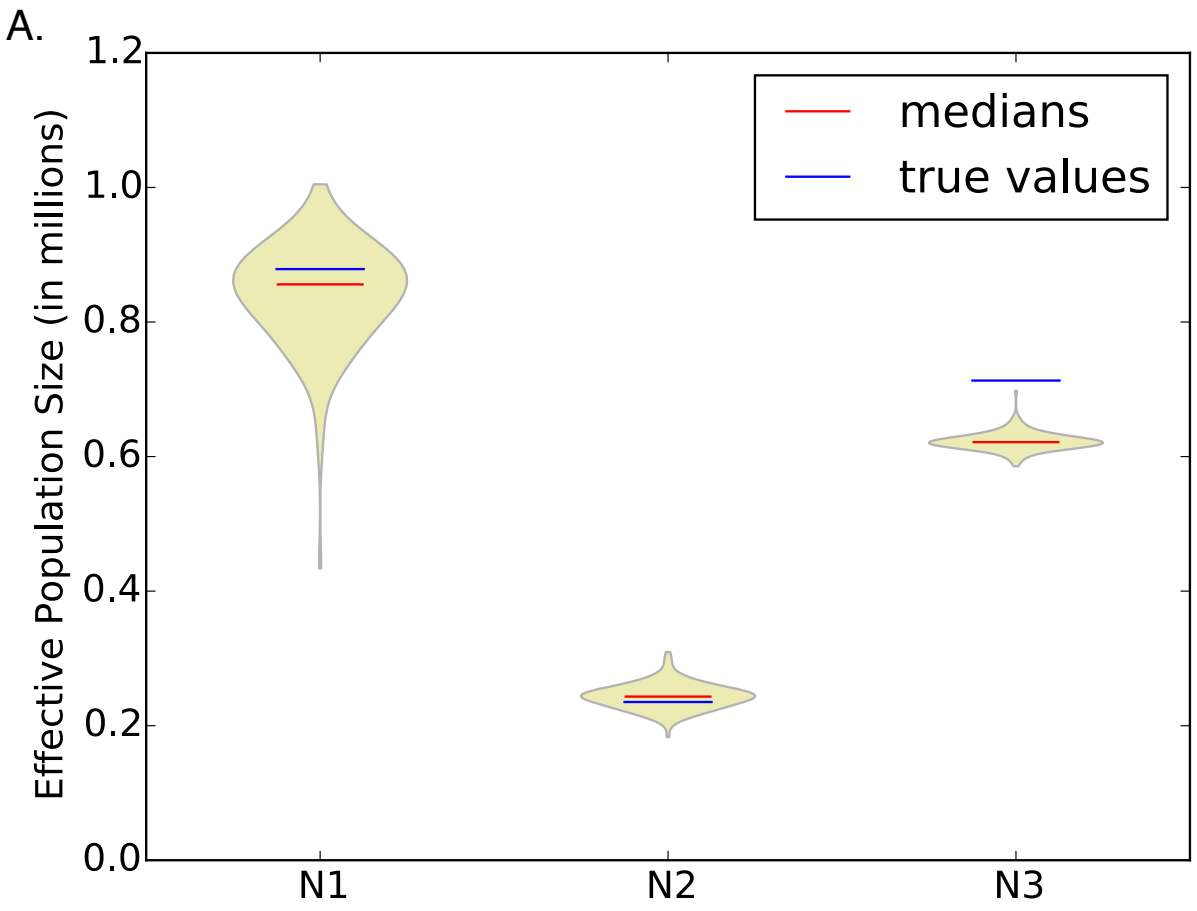
A deep learning method for population genetics



Implementation of evoNet (Java)



Population size accuracy



Population size results for an example simulated dataset.

Natural selection accuracy

Confusion Matrix

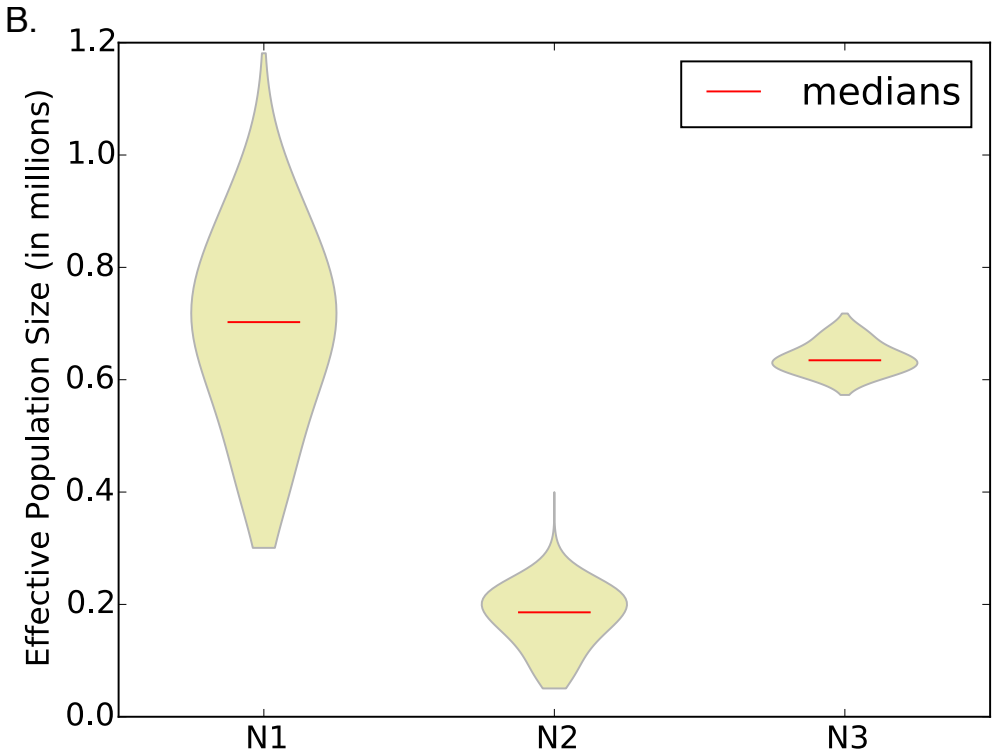
	Called Class			
True Class	Neutral	Hard Sweep	Soft Sweep	Balancing
Neutral	**0.9995**	0.0002	0.0003	0.0000
Hard Sweep	0.1434	**0.8333**	0.0032	0.0201
Soft Sweep	0.0096	0.0010	**0.9891**	0.0003
Balancing	0.0301	0.0356	0.0056	**0.9287**

Overall accuracy: 93.8%

With and without unsupervised pretraining

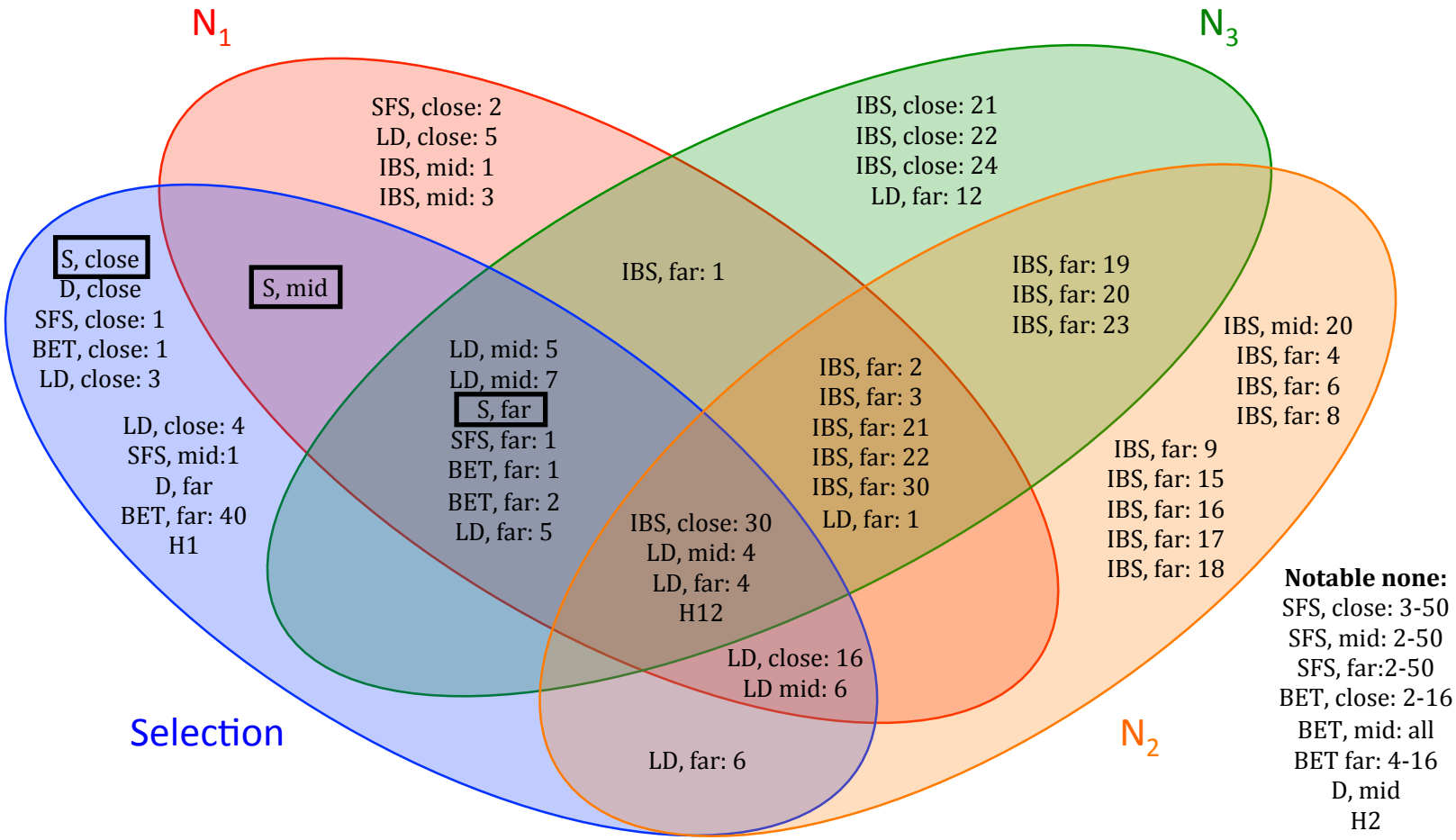
	Called Class			
True Class	Neutral	Hard Sweep	Soft Sweep	Balancing
	Random Initialization			
Neutral	**1.000**	0.000	0.000	0.000
Hard Sweep	**0.978**	0.007	0.000	0.015
Soft Sweep	**1.000**	0.000	0.000	0.000
Balancing	**1.000**	0.000	0.000	0.000
	Autoencoder Initialization			
Neutral	**1.000**	0.000	0.000	0.000
Hard Sweep	0.145	**0.831**	0.004	0.021
Soft Sweep	0.011	0.001	**0.987**	0.000
Balancing	0.030	0.028	0.001	**0.941**

Results on real data



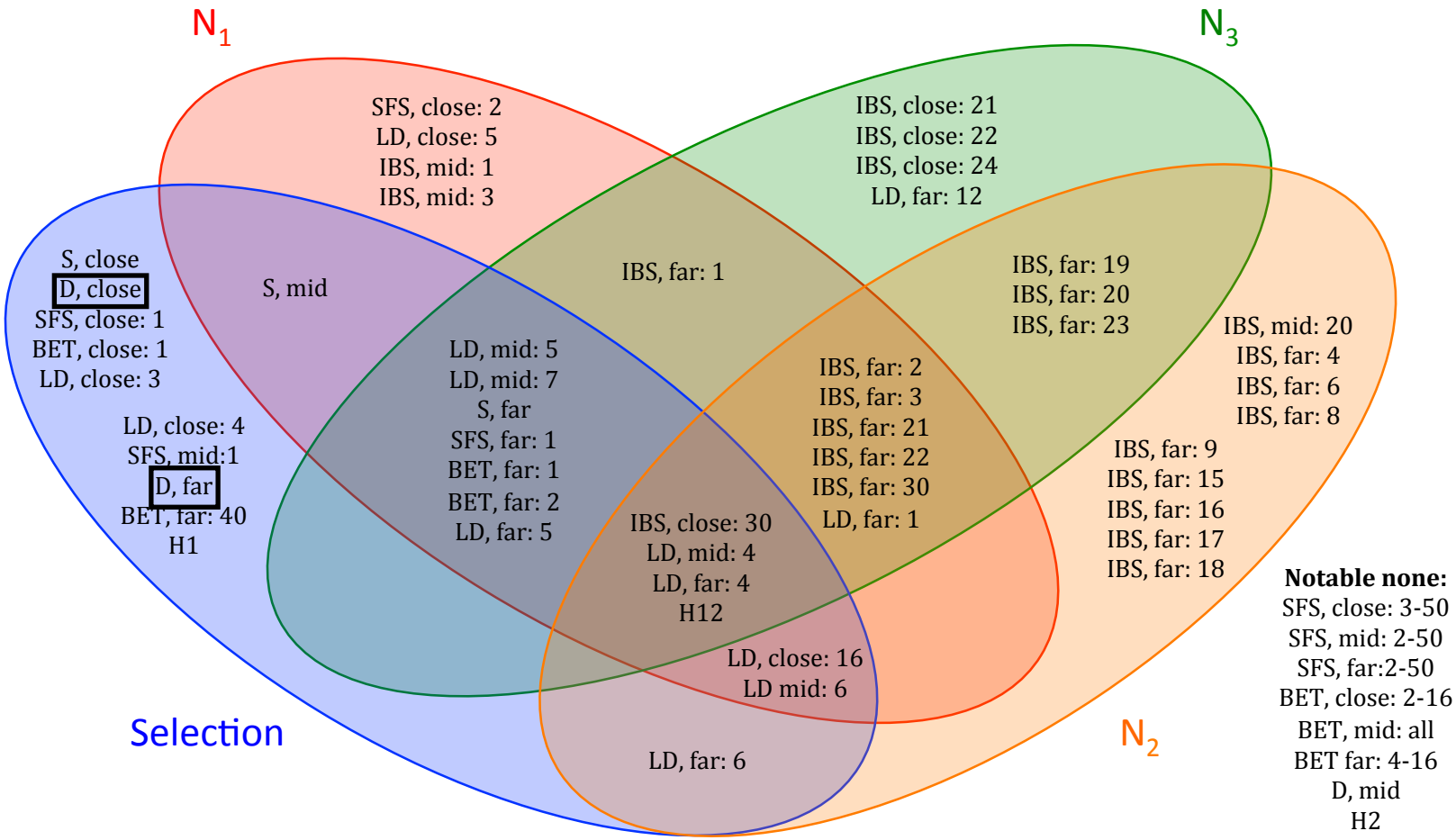
Population size history results for real *Drosophila* (fruit fly) data.

Feature selection: "best" statistics



S = number of segregating sites

Feature selection: "best" statistics



D = Tajima's D

Other methods that use summary statistics: SVM

SVM for natural selection

- Input features: SFS
- Output: selection or no selection in a gene
- Method: SVM for classification

Learning Natural Selection from the Site Frequency Spectrum

Roy Ronen,[*,1] Nitin Udpa,[*] Eran Halperin,[†] and Vineet Bafna[‡]

SVM for natural selection

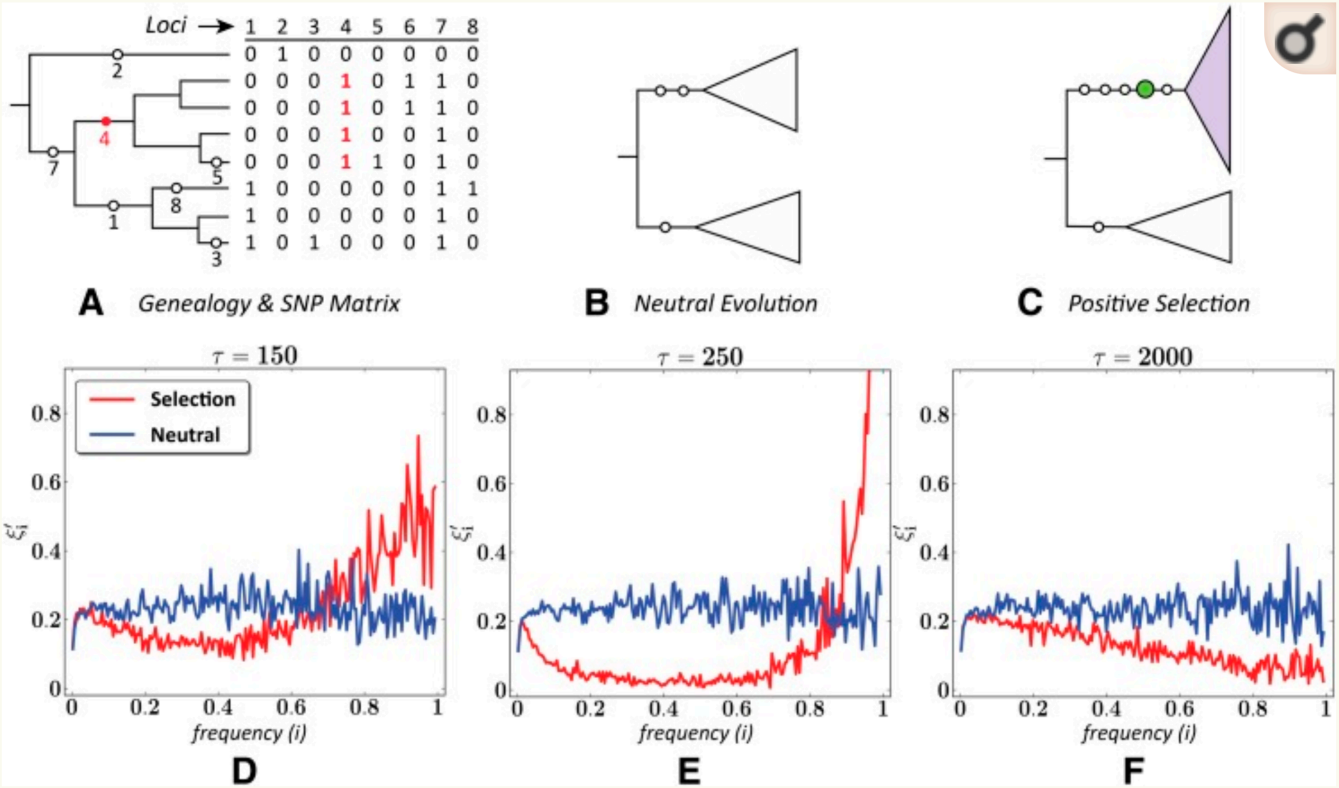


Figure 1

Impact of a selective sweep on the scaled SFS. (A) The genealogy of eight chromosomes with eight polymorphic sites falling on different branches, and the corresponding SNP matrix. (B) Two populations diverged from a source population under neutral evolution, or (C) with one under selection. (D–F) The mean scaled SFS of 500 simulated samples from populations evolving neutrally or under selection ($s = 0.08$), sampled at $\tau = 150$ (D), 250 (E), and 2000 (F) generations under selection (see *Methods* for simulation details).

SVM for natural selection

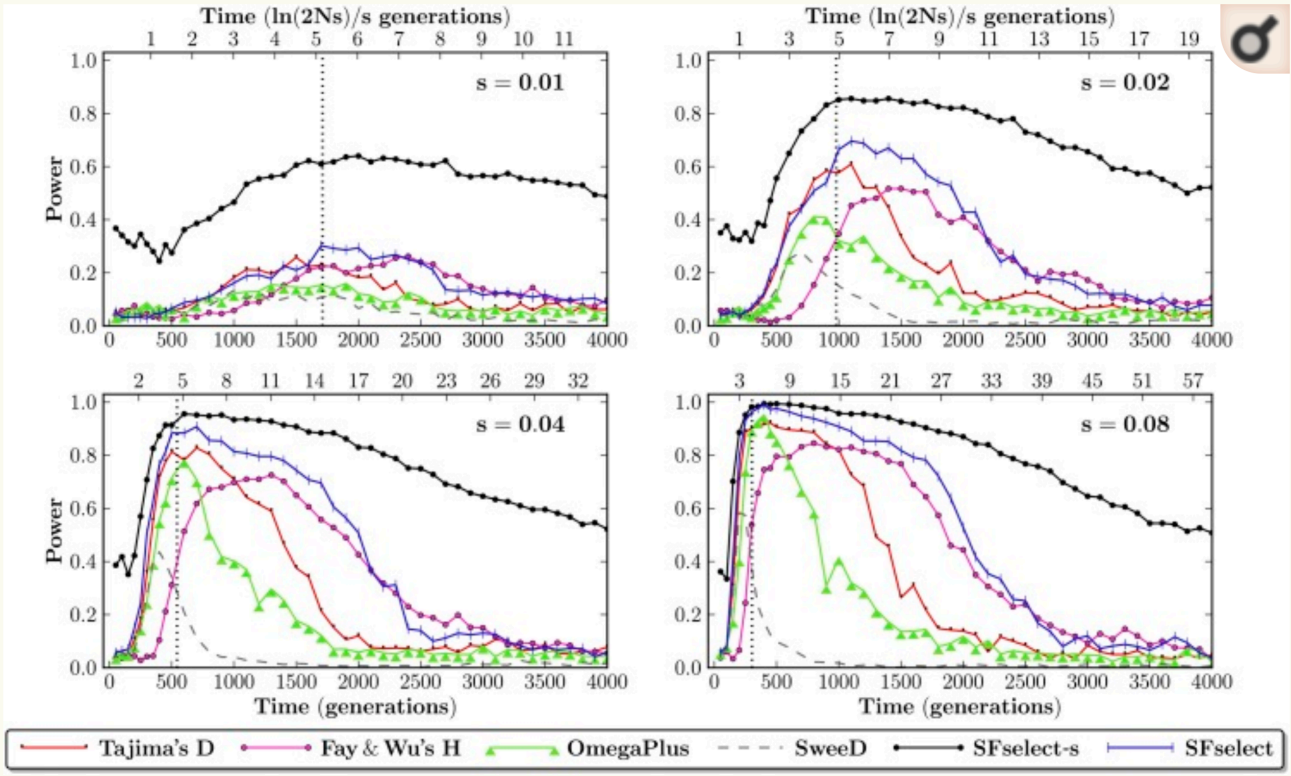
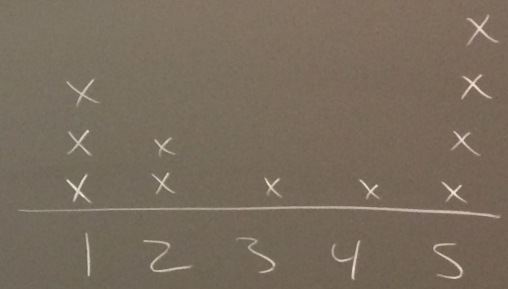


Figure 3

Power (5% FPR) of the SVM test compared to other single-population tests of neutrality. Shown for 200 data sets representing selective sweeps with selective coefficients $s \in [0.01, 0.08]$, sampled at $\tau \in [0, 4000]$ generations under selection. *SFselect-s* (black) assumes knowledge of (τ, s), while *SFselect* (blue) assumes no prior knowledge of these parameters. Time is shown in generations (bottom axes), and $\ln(2Ns)/s$ generations (top axes). Dotted vertical lines show the mean time to fixation of the beneficial allele, which occurs at ≈ 5 $\ln(2Ns)/s$ generations.

$$\text{power} = 1 - \underbrace{P\left(\text{said: no selection} \mid \text{truth: selection}\right)}_{\text{type II error}}$$



$n = 6$

oduct

1

Other methods that use summary statistics: ABC

Previous work on likelihood-free inference

Big question: How can we infer population genetic parameters when we don't know the genealogy?

One answer: Approximate Bayesian Computation (ABC)

$(\vec{z})$ parameters of interest

data $(\vec{x})$

prior

posterior

$$p(\theta \mid D) = \frac{p(\theta)\, p(D \mid \theta)}{p(D)}$$

selection
mutation rate
recomb rate

ABC

$\hat{\theta} \sim p(\theta)$

Simulate $\hat{D} \sim M_{\hat{\theta}}$ ← model (coalescent)

compare $\quad \mathrm{dist}(D, \hat{D}) < \varepsilon$

$\boxed{\text{keep } \hat{D}}$

scent)

$< \varepsilon$

real vs. simulated

summary statistics

$$\Rightarrow \text{dist}(S(D), S(\hat{D})) < \varepsilon$$

θ^* = avg(all $\hat{\theta}$ that correspond to $\hat{D}$ that I kept)

param for real data

Lasagna Failure

Previous work on likelihood-free inference

Big question: How can we infer population genetic parameters when we don't know the genealogy?

One answer: Approximate Bayesian Computation (ABC)

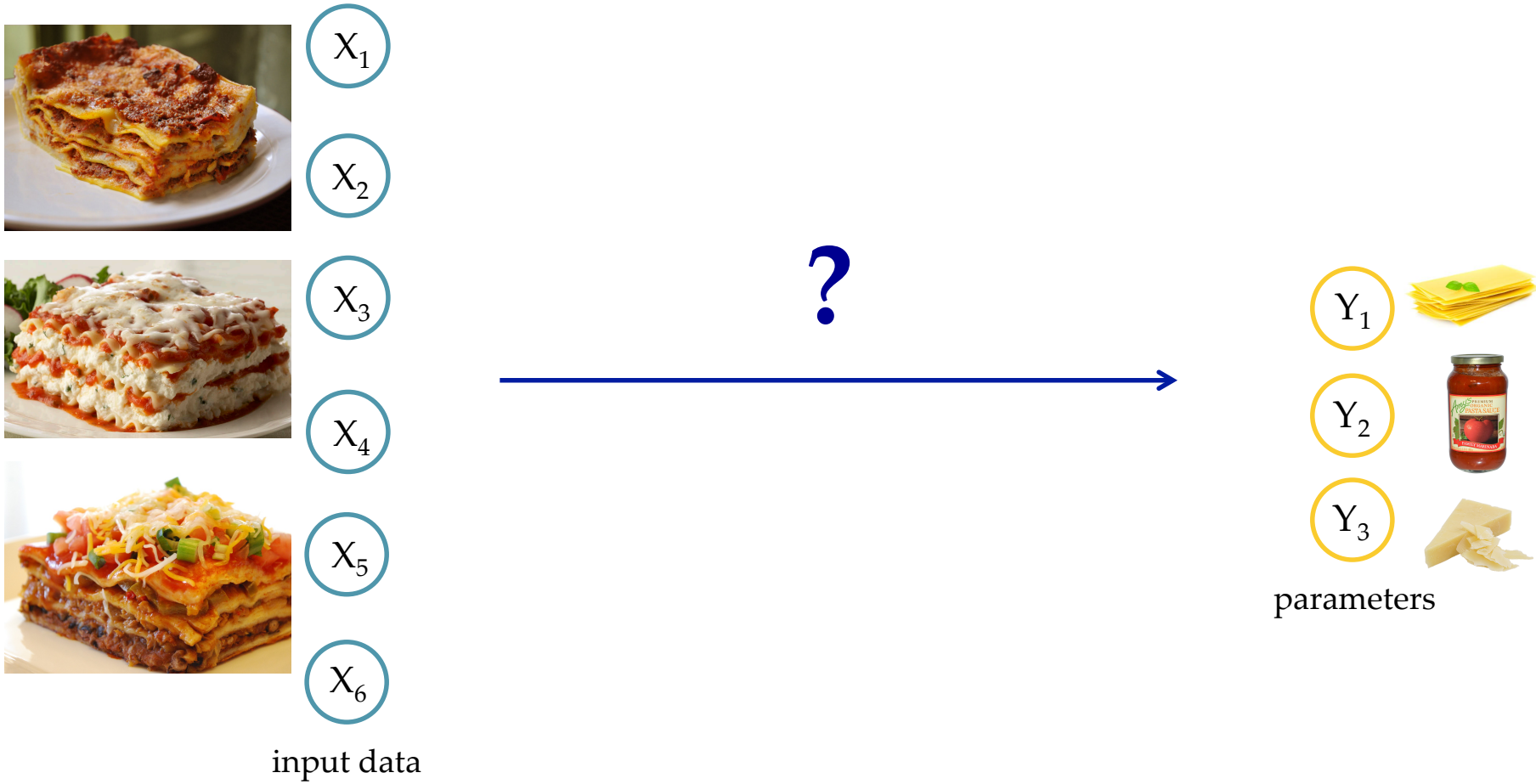
Advantages of ABC:

- easy to use
- always gives an answer

Disadvantages of ABC:

- rejection method
- hard to interpret
- "curse of dimensionality"
- distance metric on datasets

Another answer: machine learning



Moving away from summary statistics: convolutional neural networks

Major drawback of this deep learning method

Requires data to be compressed into "expert" summary statistics

- Best statistics will change for each application
- No matter how many are used, still losing information
- Bottleneck in data analysis

Problems using the raw data

L

n

```
000000000000100000000001000000
000000100000000000000000000000
000000100000000000000001000000
000000000000100000000001000000
000000100000000000000001000000
000000000000000000000001000000
```

Dataset 1

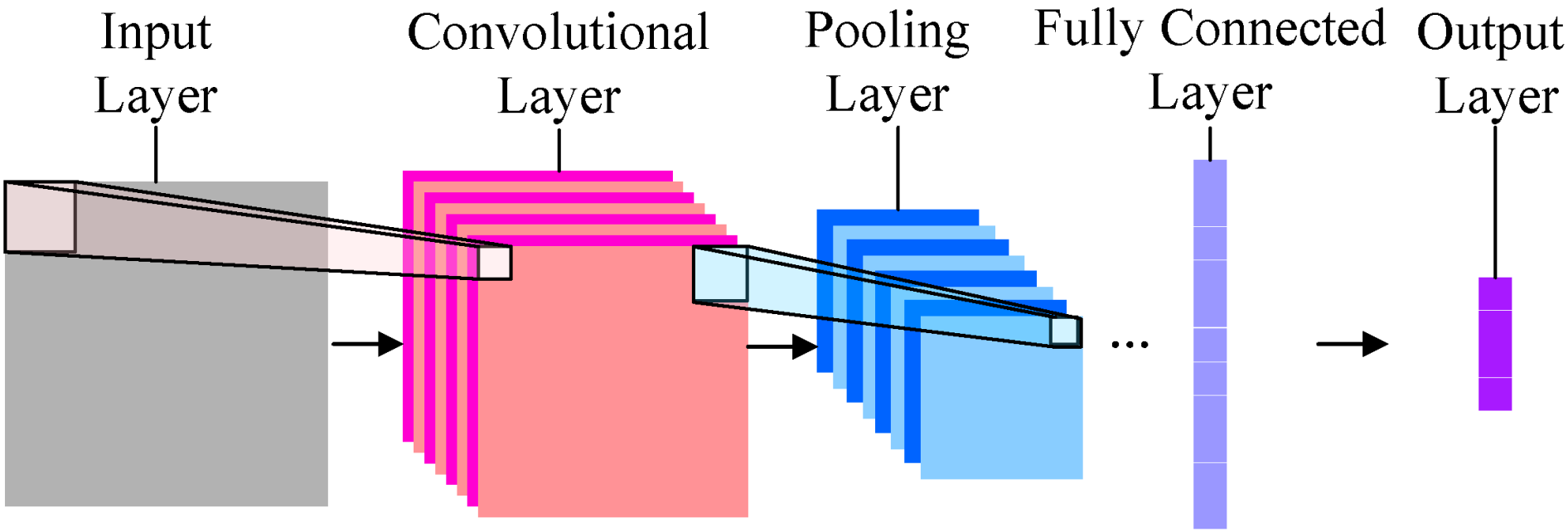
```
010000000000000000000000000000
010000000000000000100000000000
000000000100000000100000000000
010000000000000000000000000000
010000000000000000100000000000
000000000000000000100000000000
```

Dataset 2

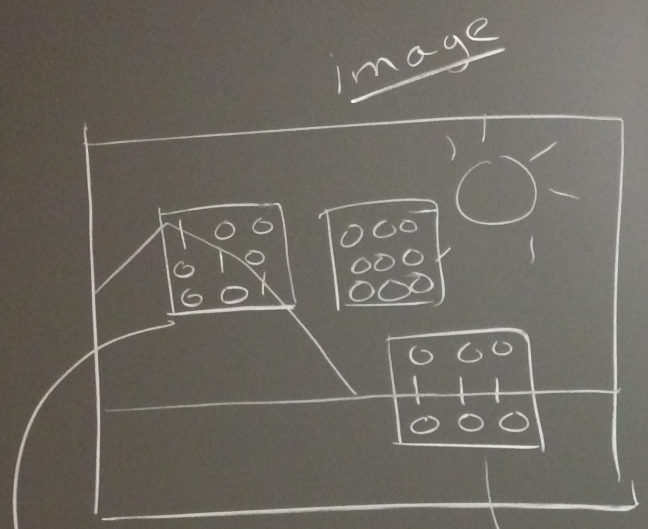
Issues:

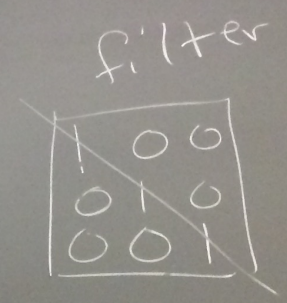
1. $n = 100,\ L = 100,000$ (data size)
2. Dataset 1 and Dataset 2 generated under the same model

Convolutional Neural Networks



NIRFaceNet: A Convolutional Neural Network for Near-Infrared Face Identification (2016)





power

dot product

3 1

Convolutional neural networks in TensorFlow

sliding window with many filters

```
010000000000000000000000000000
010000000000000000100000000000
000000000100000000100000000000
010000000000000000000000000000
010000000000000000100000000000
000000000000000000100000000000
```

Architecture summary:

- Two convolutional layers
- One fully connected layer
- ReLU activation function
- Max pooling
- Batch training

More info about CNNs: Stanford course
http://cs231n.stanford.edu/

Convolutional neural networks in TensorFlow

sliding window with many filters

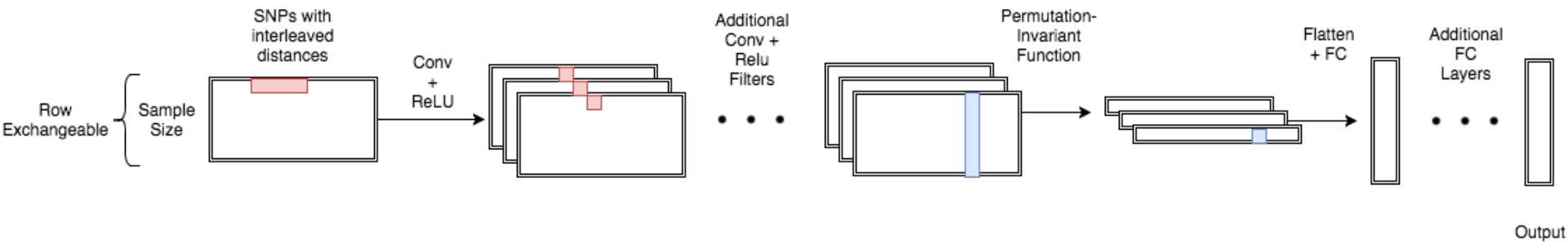
```
010000000000000000000000000000
010000000000000000100000000000
000000000100000000100000000000
010000000000000000000000000000
010000000000000000100000000000
000000000000000000100000000000
```

Architecture summary:

- Two convolutional layers
- One fully connected layer
- ReLU activation function
- Max pooling
- Batch training

Test application: recombination

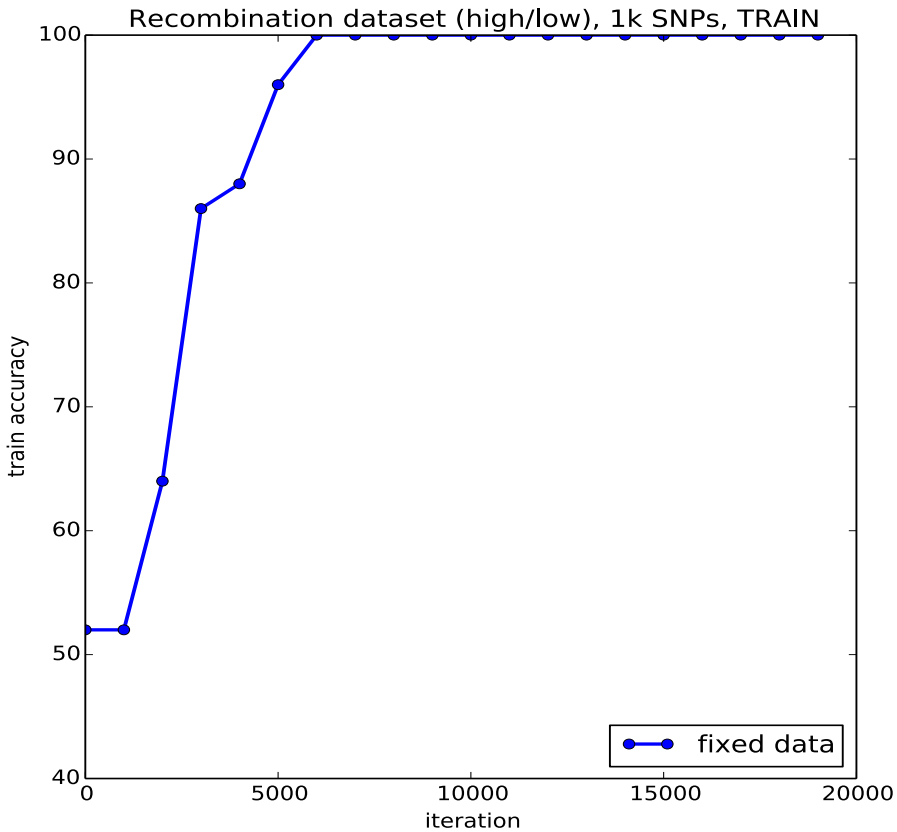
Neural network diagram



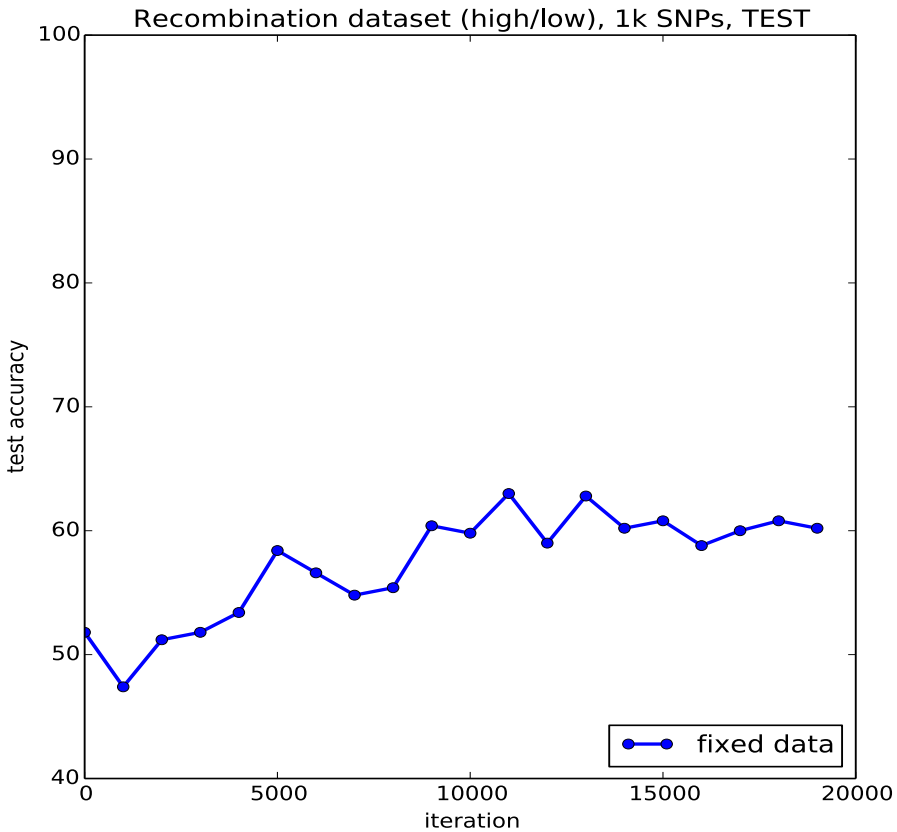
A Likelihood-Free Inference Framework for Population Genetic Data using Exchangeable Neural Networks
Jeffrey Chan, Valerio Perrone, Jeffrey P. Spence, Paul A. Jenkins, **Sara Mathieson**, Yun S. Song
arXiv preprint, February 2018

Preliminary results: fixed dataset

Training Data

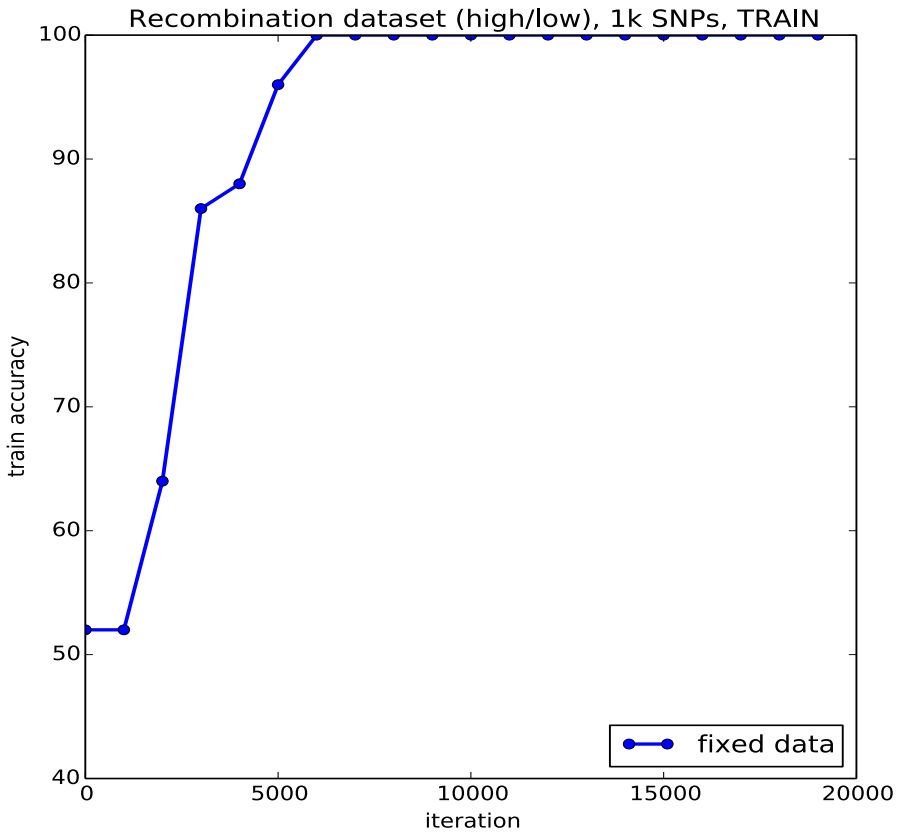


Test Data

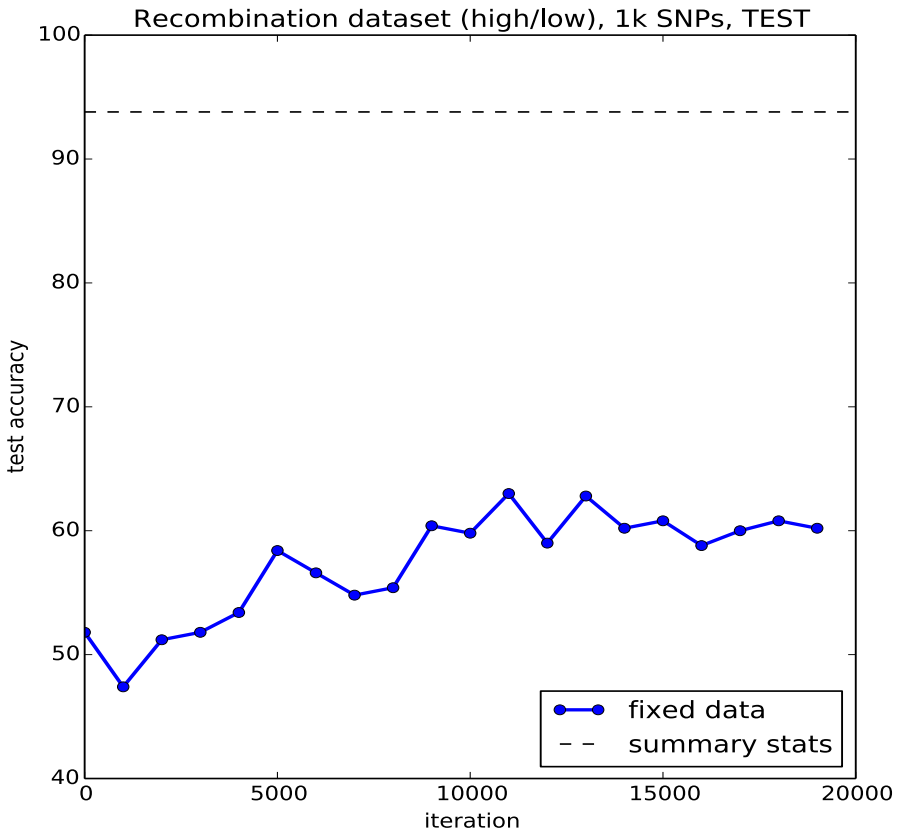


Preliminary results: fixed dataset

Training Data

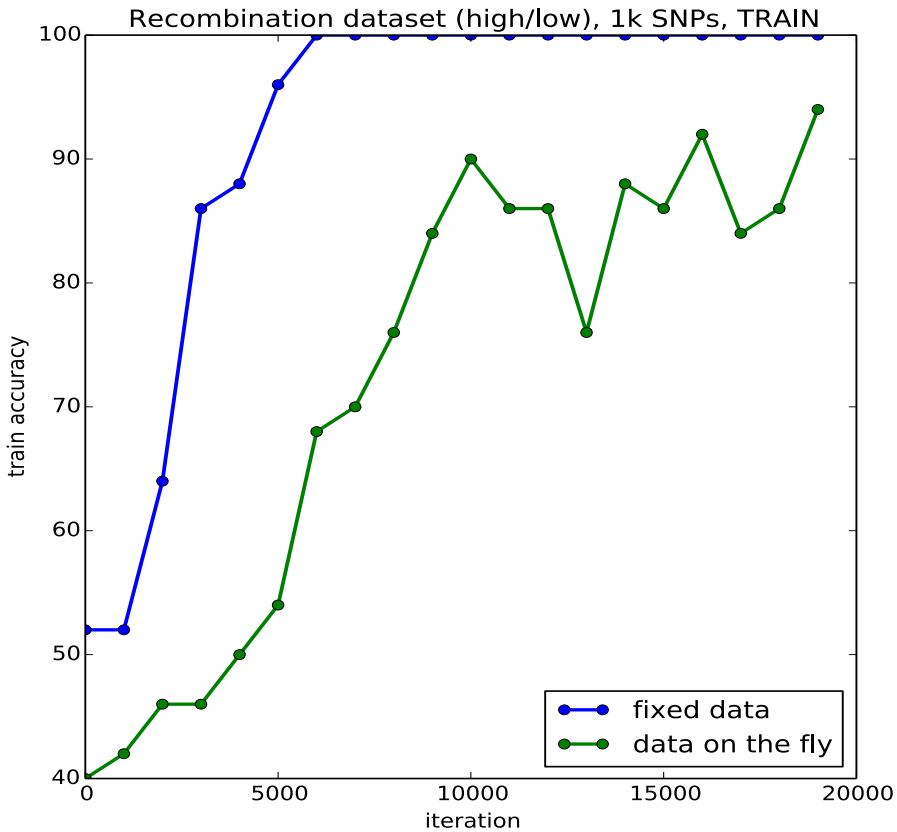


Test Data

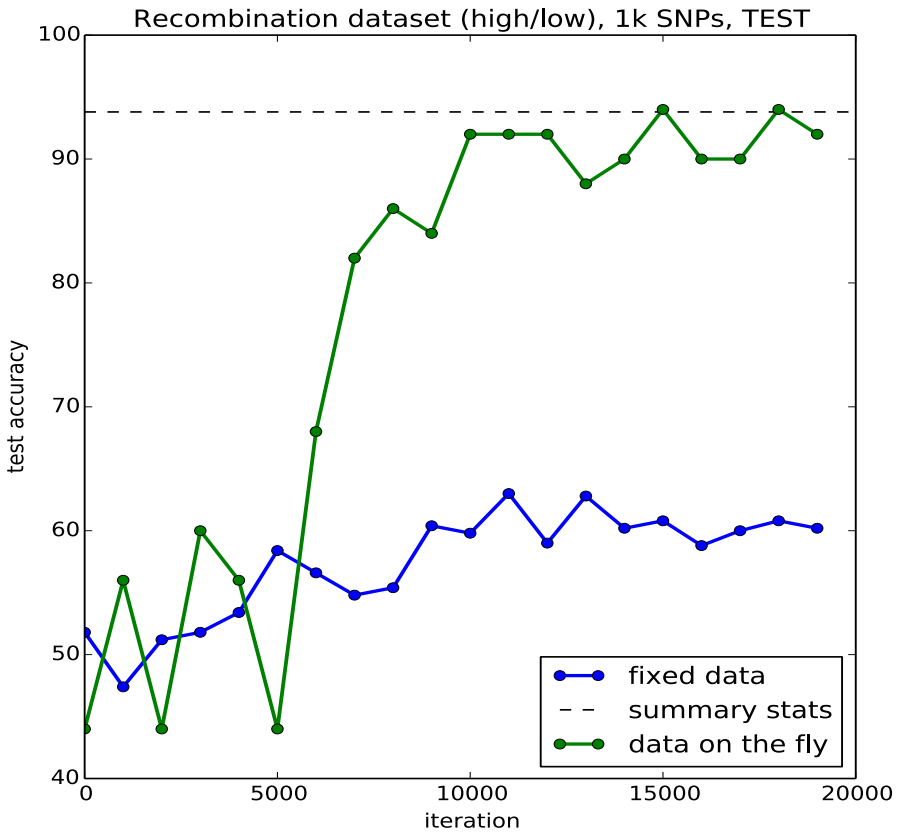


Preliminary results: simulating data during training

Training Data



Test Data



Other biological applications

- Transcription-factor binding site prediction
- Sequence clustering
- Predicting disease status from image scans
- Other applications of HMMs (beyond PSMC)