

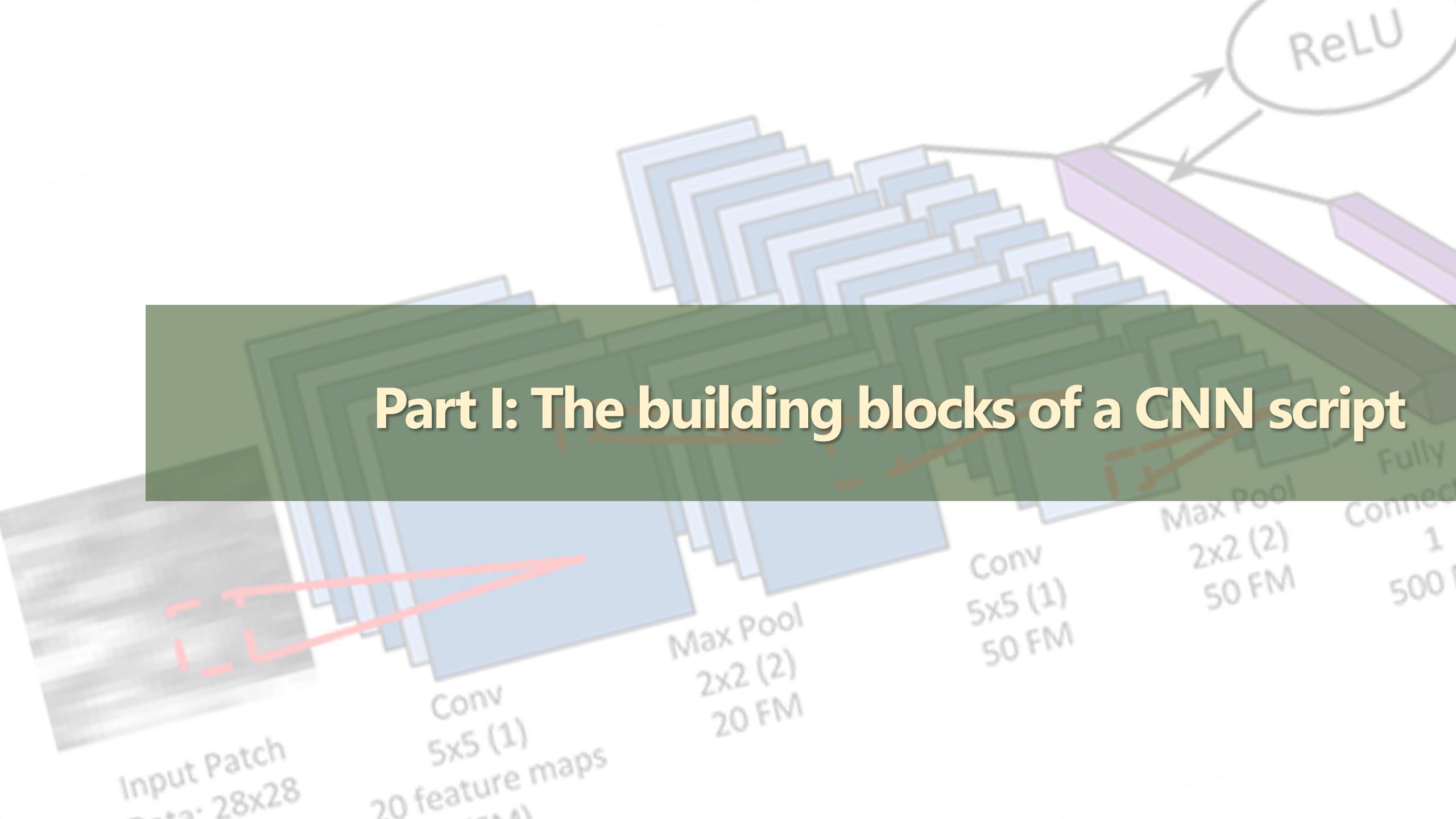
Course goals

- Conceive and simulate genetic data under competing demographic scenarios 
- Understand deep learning background and how a CNN works 
- Simulate genomic regions with selective sweeps and use CNN to detect such regions on real genomes 
- How to use deep learning to compare demographic scenarios

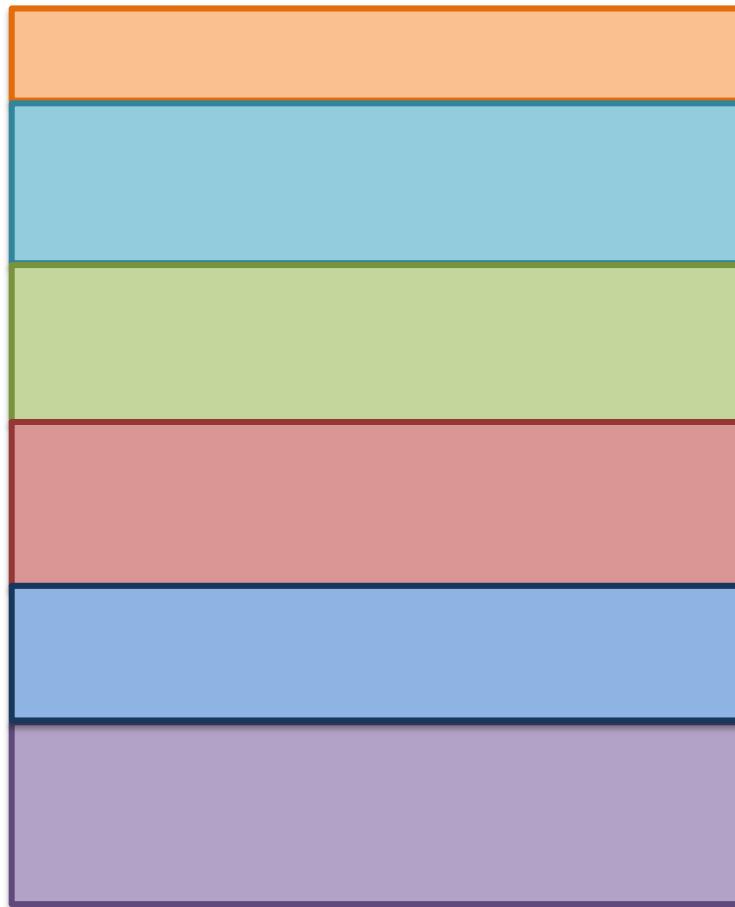
Using Convolutional neural Networks (CNNs) for demographic inference

Program

Part I: The building blocks of a CNN script



CNN Script



required python modules.

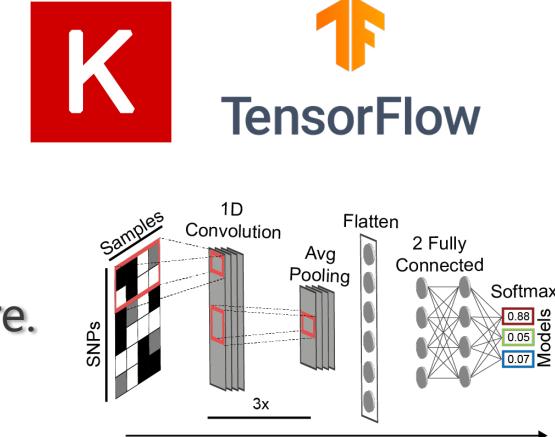
Define the CNN architecture.

Load and process the training data.

Train the network.

Load the test data and perform cross-validation.

Predict the most likely model for the empirical data



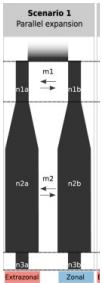
CNN Script

Inputs:

Scenario 1

Samples

-1	-1	1	0
1	-1	1	1
-1	0	1	-1
-1	1	1	1
1	1	-1	0
0	1	-1	-1
-1	0	1	-1
-1	1	1	1
1	1	-1	0
0	1	-1	-1



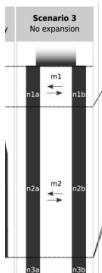
Scenario 2

N_o
of simulations



Scenario 3

-1	-1	1	0
1	-1	1	1
-1	0	1	-1
-1	1	1	1
1	1	-1	0
0	1	-1	-1
-1	0	1	-1
-1	1	1	1
1	1	-1	0
0	1	-1	-1

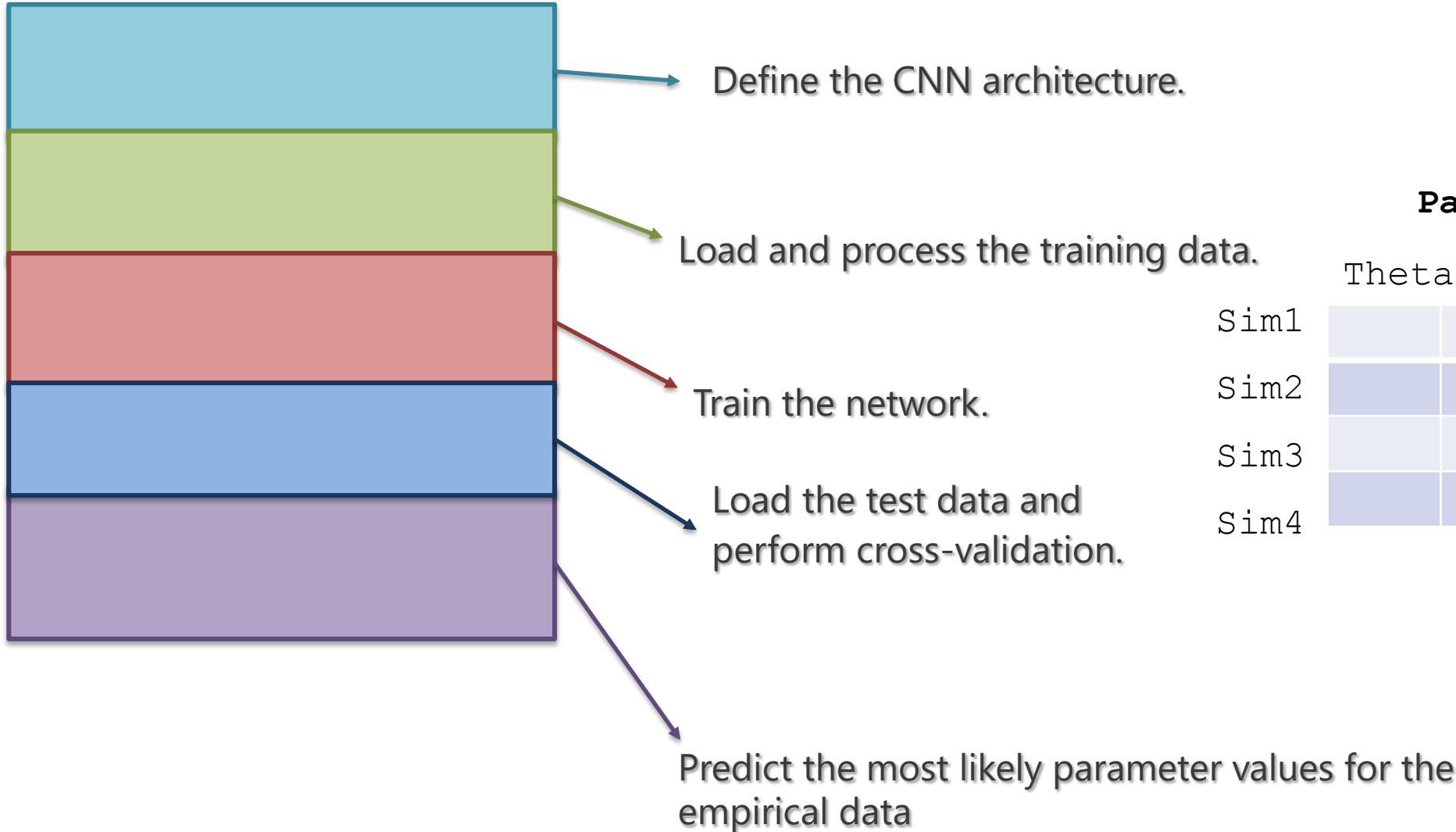


Parameters

Theta T1 T2 T3 Ne

Sim1				
Sim2				
Sim3				
Sim4				

CNN Script



Parameters

Theta T1 T2 T3 Ne

Sim1

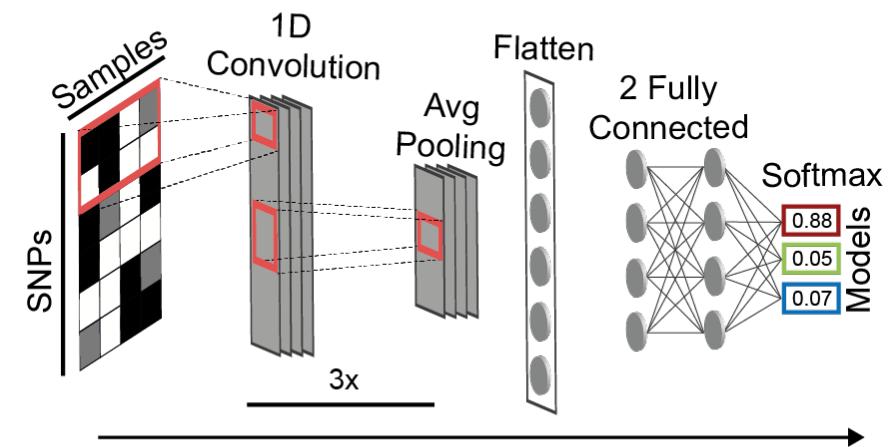
Sim2

Sim3

Sim4

Theta	T1	T2	T3	Ne
Sim1				
Sim2				
Sim3				
Sim4				

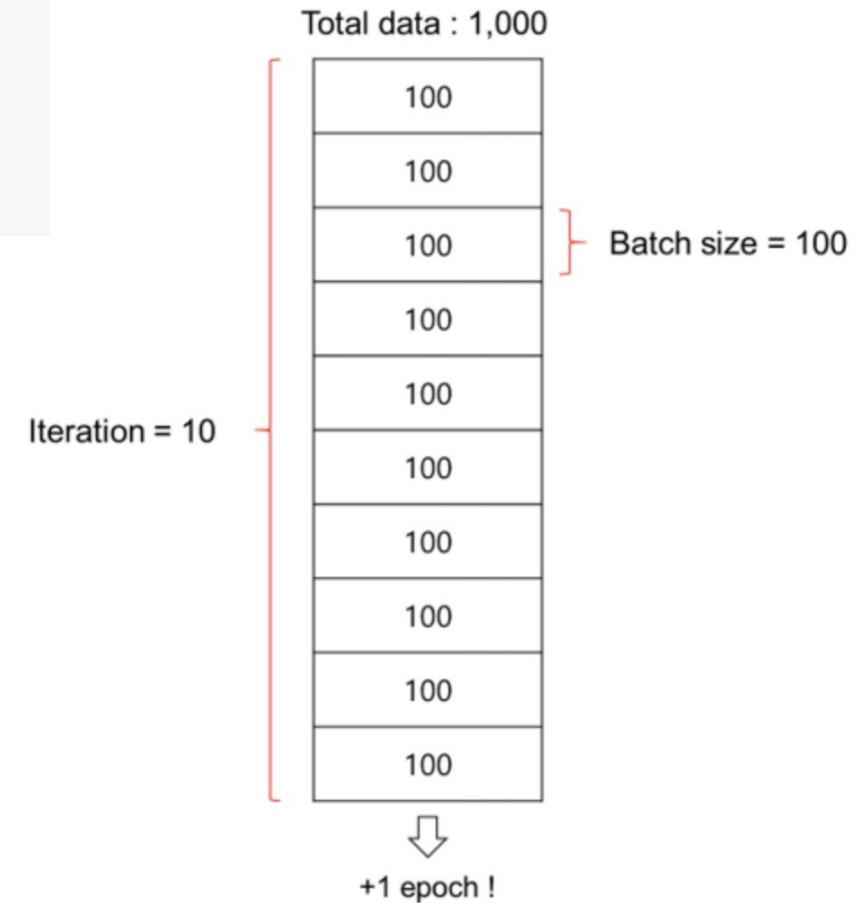
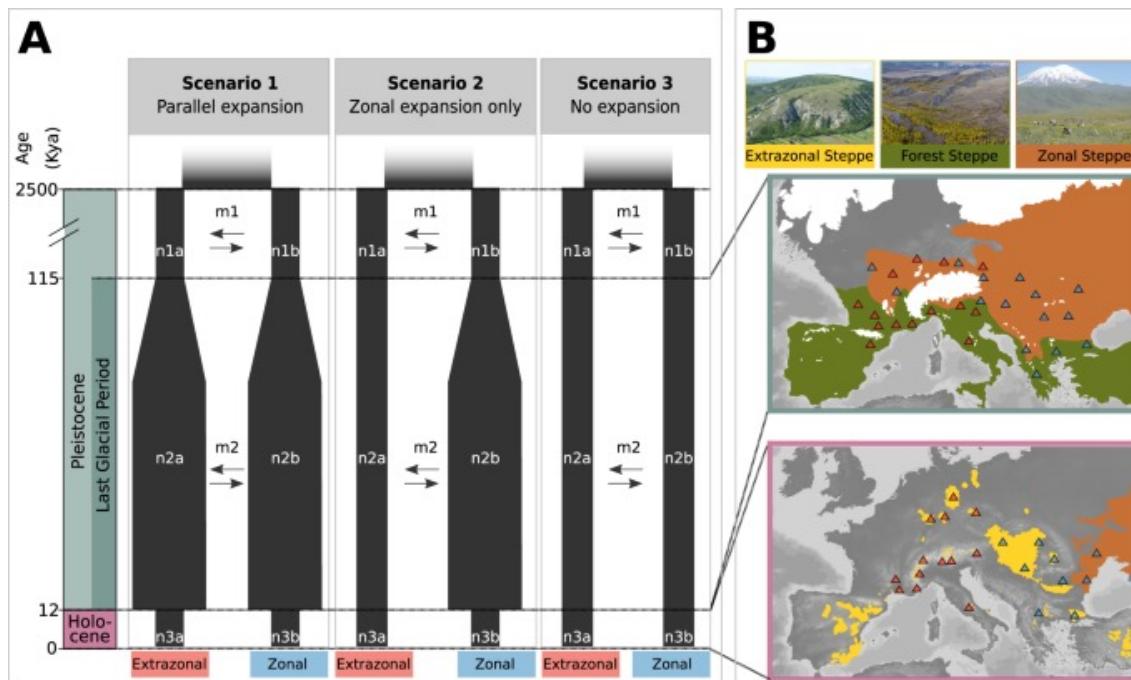
- **Practical Exercise 1:**
- Go through Section 1 of the script and try to recognize all the elements of the network. Do you remember the function of each of those elements? Remember that you can add annotations to the code using # and add information that might help you when you get back to the script in the future.
- Now run all the cells until you reach the end of section 2. Your network will be training, so now we will have some time to discuss and do a quick review on the CNN elements.



CNN Script

```
# Define parameters for the CNN run.
batch_size = 200
### how much interations to train the network
epochs = 100

###n of models
num_classes = 3
```

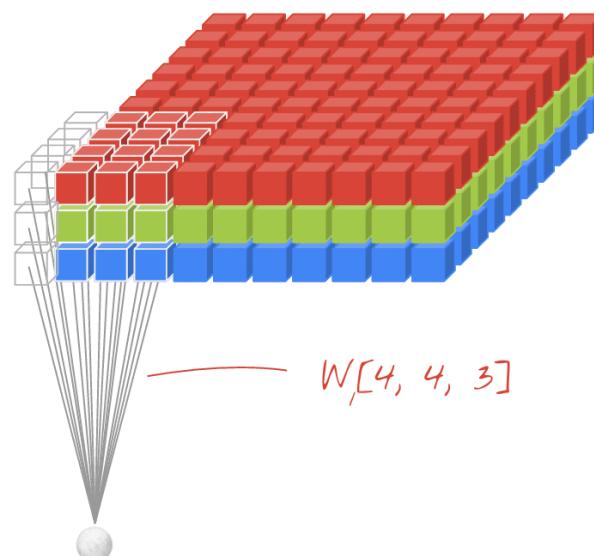


<https://jerryan.medium.com/batch-size-a15958708a6>

Kirschner et al. (2022) *Nat Comm*

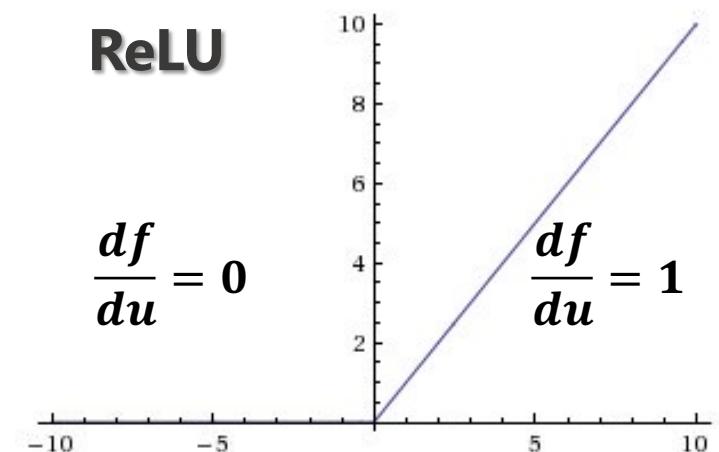
CNN Script

```
# Define the CNN architecture.
def create_cnn(xtest):
    inputShape = (xtest.shape[1], xtest.shape[2])
    ## image size. images need to have EXACTLY the same size
    inputs = Input(shape=inputShape)
    x = inputs
    ## 1D convolution - less computational intensive and also treats snps as independent;
    x = Conv1D(250, kernel_size=2, activation='relu',input_shape=(xtest.shape[1], xtest.shape[2]))(x)
    ### Enables the network to learn more complex features / shapes.
    x = AveragePooling1D(pool_size=2)(x)
    x = BatchNormalization()(x)
```



ReLU

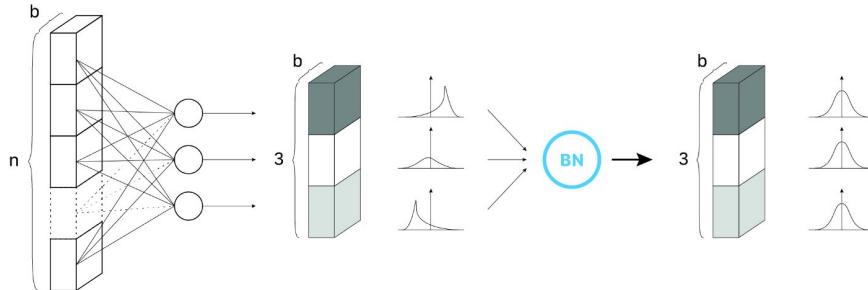
$$\frac{df}{du} = 0$$



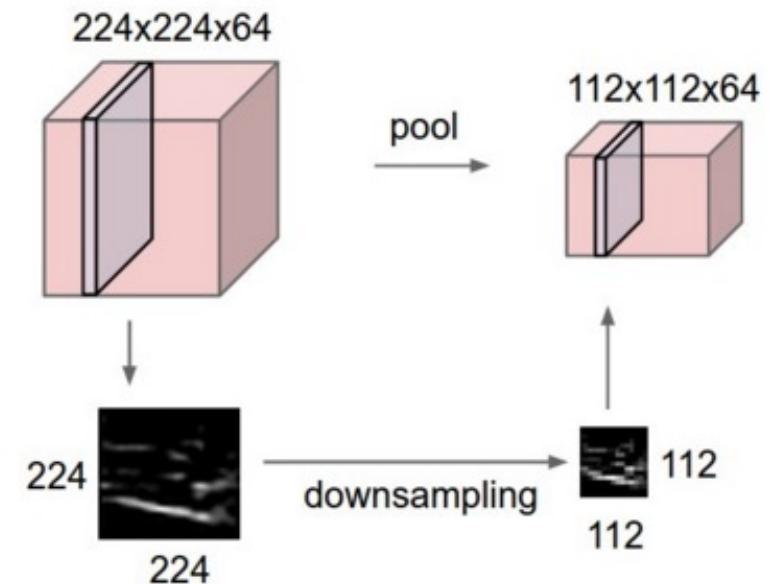
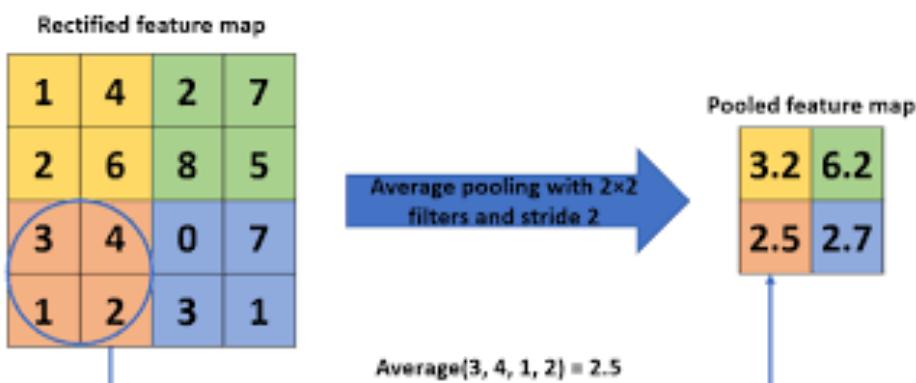
<https://www.kaggle.com/code/dansbecker/rectified-linear-units-relu-in-deep-learning/notebook>

CNN Script

```
# Define the CNN architecture.
def create_cnn(xtest):
    inputShape = (xtest.shape[1], xtest.shape[2])
    ## image size. images need to have EXACTLY the same size
    inputs = Input(shape=inputShape)
    x = inputs
    ## 1D convolution - less computational intensive and also treats snps as independent;
    x = Conv1D(250, kernel_size=2, activation='relu', input_shape=(xtest.shape[1], xtest.shape[2]))(x)
    ### Enables the network to learn more complex features / shapes.
    x = AveragePooling1D(pool_size=2)(x)
    x = BatchNormalization()(x)
```



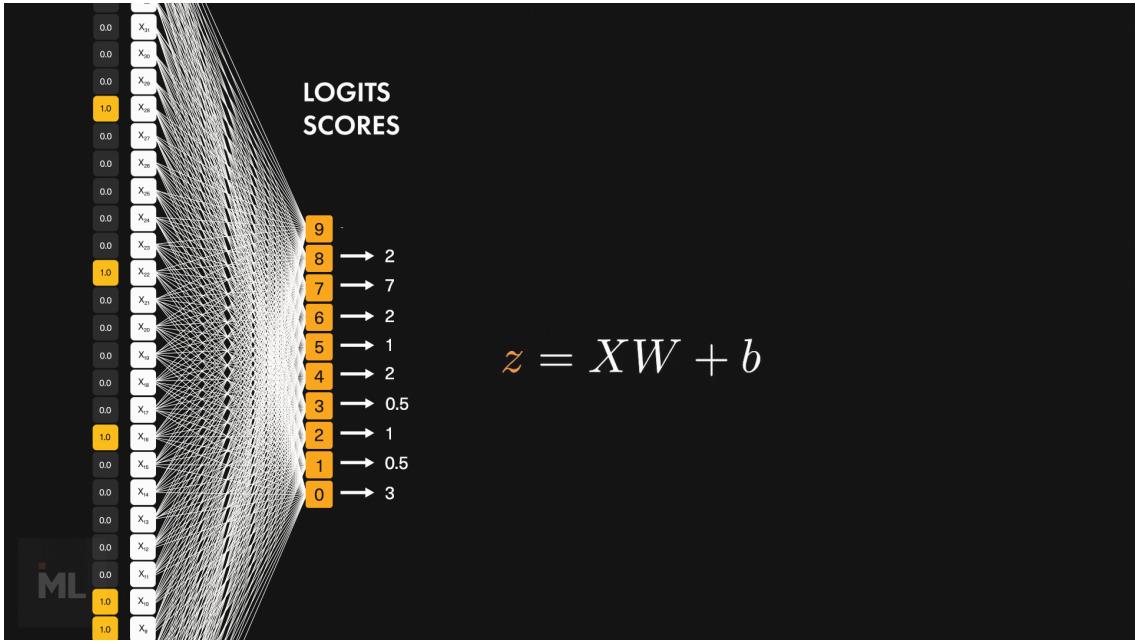
<https://towardsdatascience.com/batch-normalization-in-3-levels-of-understanding-14c2da90a338>



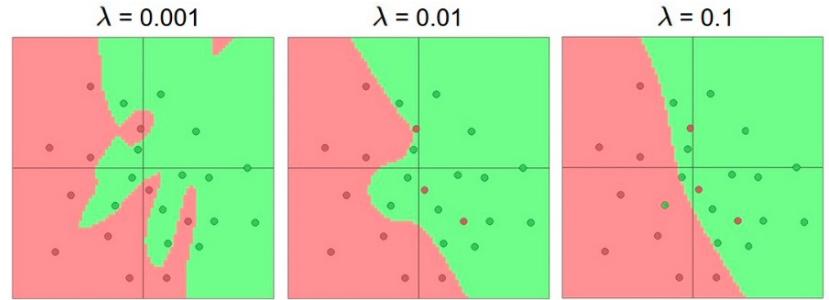
https://leonardoaraujosantos.gitbook.io/artificial-intelligence/machine_learning/deep_learning/pooling_layer

CNN Script

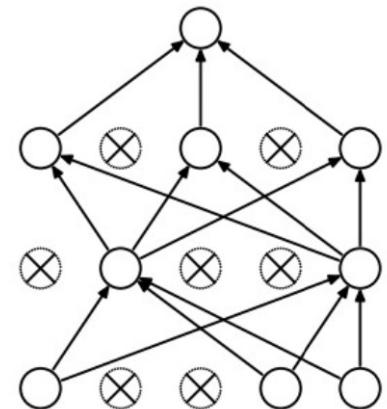
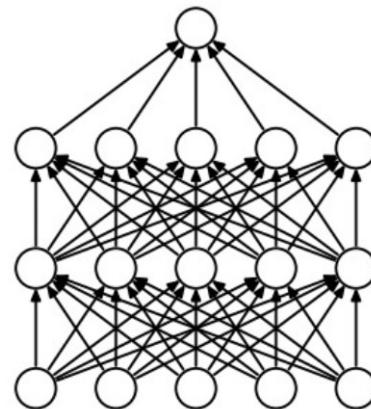
```
### Linearising the image as in the initial step. From this point on the network behaves as a Multi-Layer Perceptron
x = Flatten()(x)
x = Dense(125, activation='relu',kernel_regularizer=l2(1e-3), bias_regularizer=l2(1e-3))(x)
x = Dropout(0.5)(x)
x = Dense(125, activation='relu',kernel_regularizer=l2(1e-3), bias_regularizer=l2(1e-3))(x)
x = Dropout(0.5)(x)
x = Dense(num_classes, activation="softmax")(x)
```



Regularization



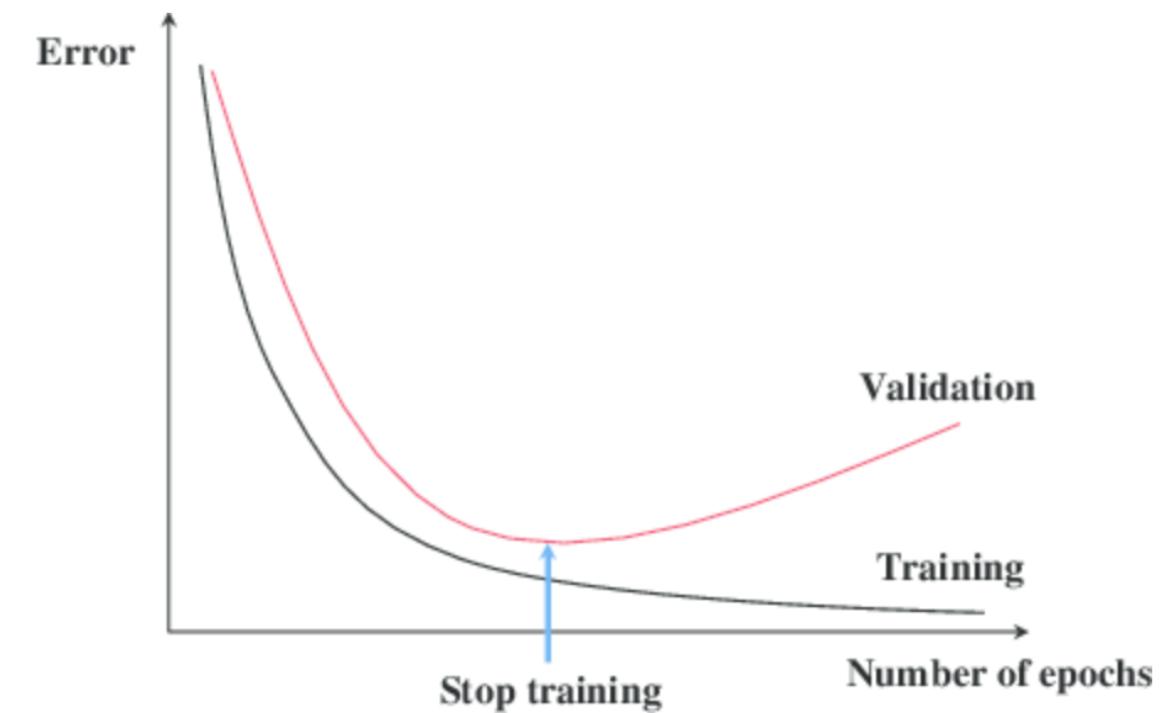
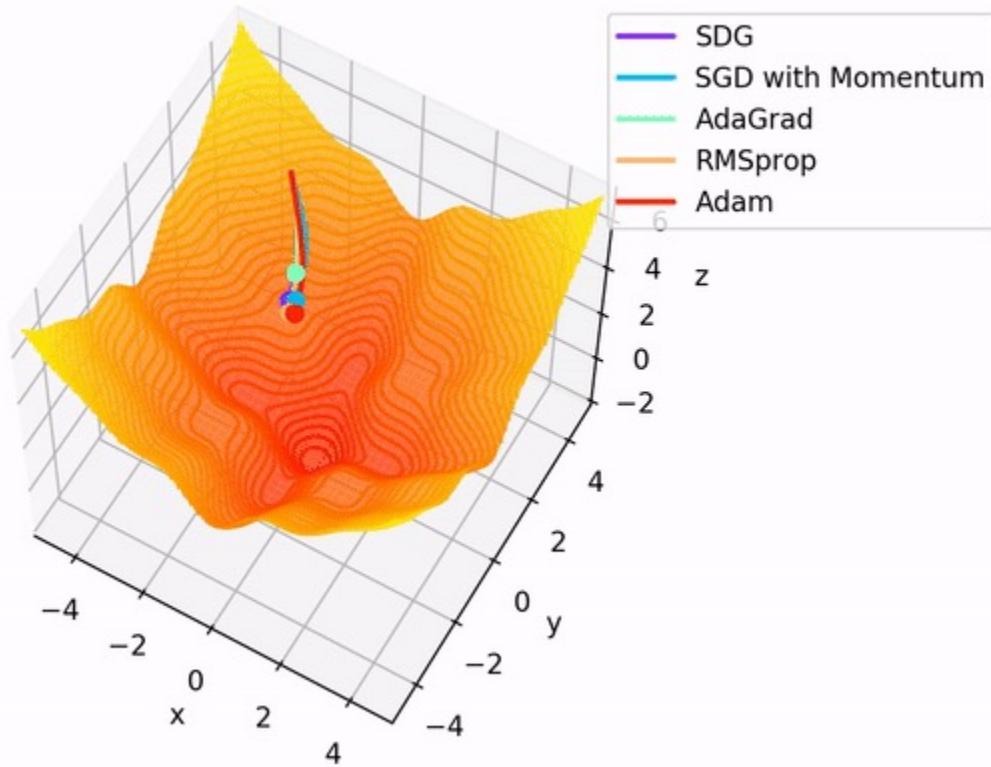
Options: L2, L1, maxnorm and dropout.



CNN Script

```
# Compile the CNN.  
model.compile(loss=keras.losses.categorical_crossentropy,  
              optimizer='Adam',  
              metrics=['accuracy'])  
  
# We will use early stopping and save the model with the best val_accuracy.  
earlyStopping = EarlyStopping(monitor='val_accuracy', patience=150, verbose=0, mode='max', restore_best_weights  
### stop training when validation increases error
```

Optimizer Comparison



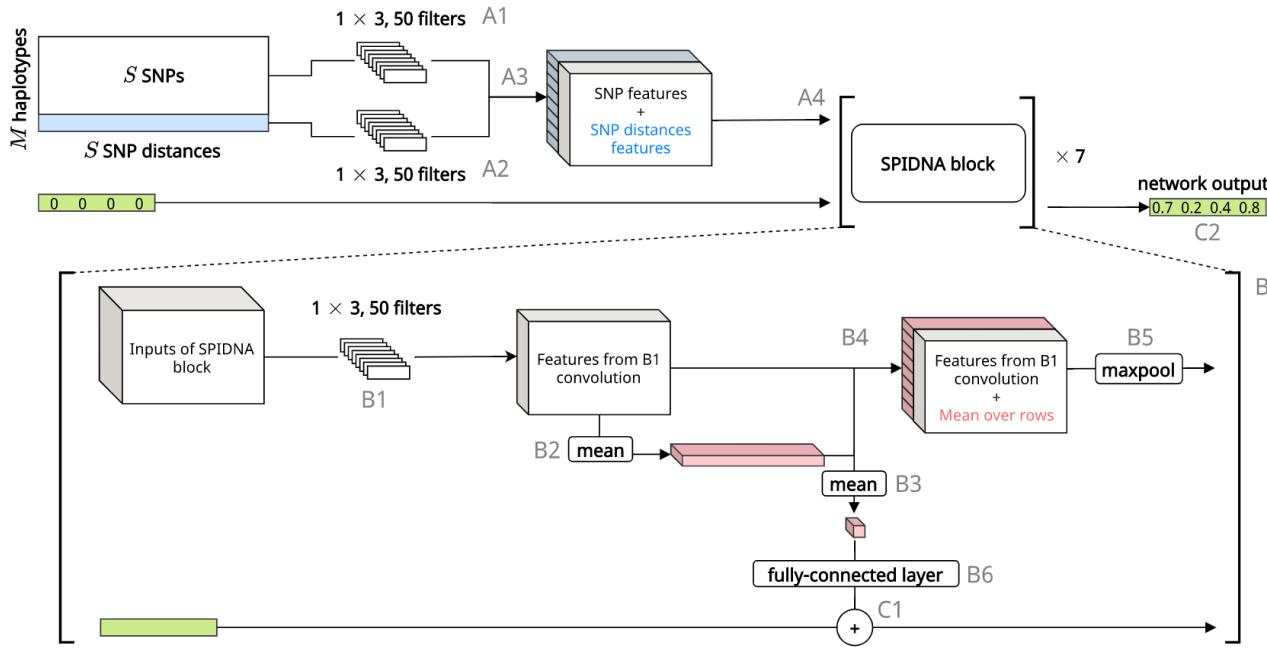
<https://towardsdatascience.com/complete-guide-to-adam-optimization-1e5f29532c3d>

<https://towardsdatascience.com/a-practical-introduction-to-early-stopping-in-machine-learning-550ac88bc8fd>

Part II: Quick overview of other applications and future perspectives.

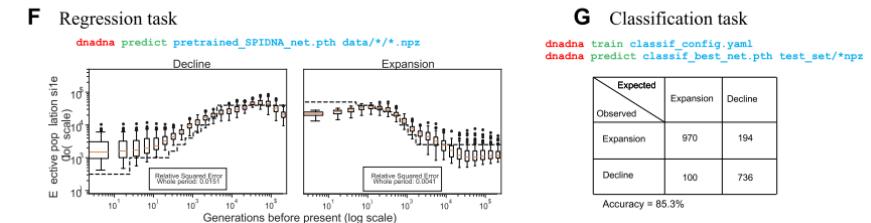
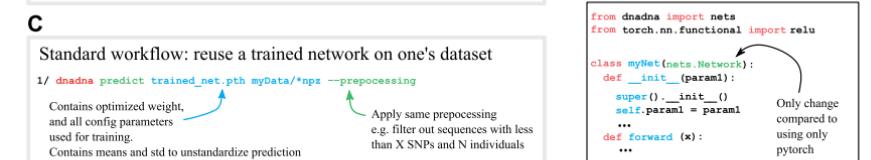
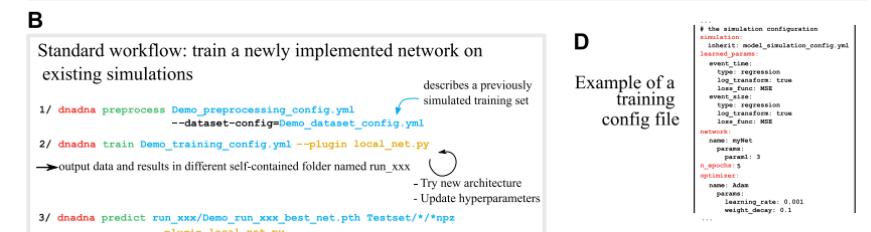
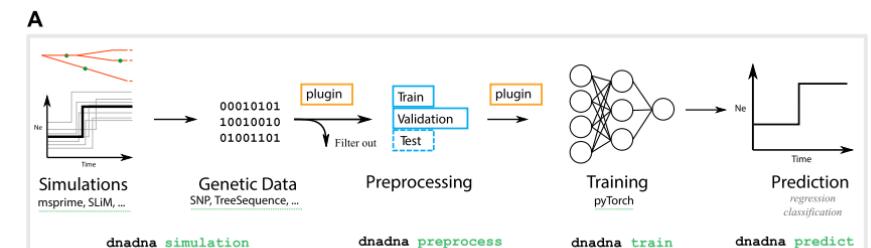
Deep learning for population size history inference: Design, comparison and combination with approximate Bayesian computation

Théophile Sanchez | Jean Cury | Guillaume Charpiat | Flora Jay



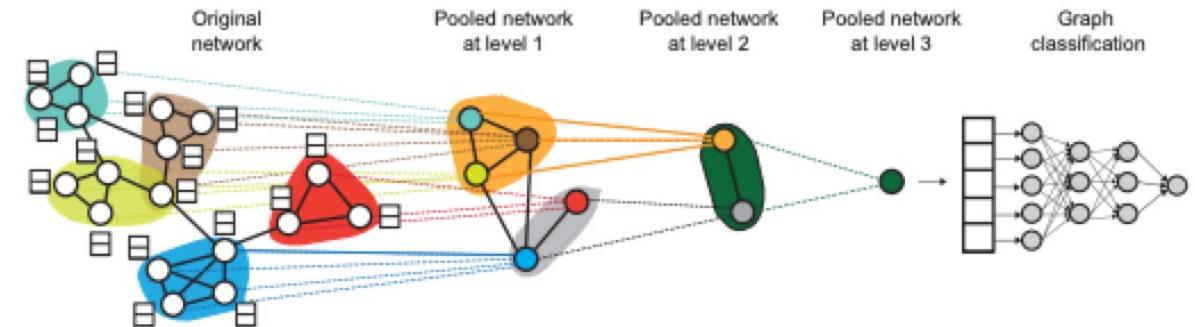
Genetics and population analysis dnadna: a deep learning framework for population genetics inference

Théophile Sanchez^{1†}, Erik Madison Bray^{1†}, Pierre Jobic^{1,2}, Jérémie Guez^{1,3}, Anne-Catherine Letournel¹, Guillaume Charpiat¹, Jean Cury ^{1,4*}‡ and Flora Jay ^{1,*†}



Simultaneous Inference of Past Demography and Selection from the Ancestral Recombination Graph under the Beta Coalescent

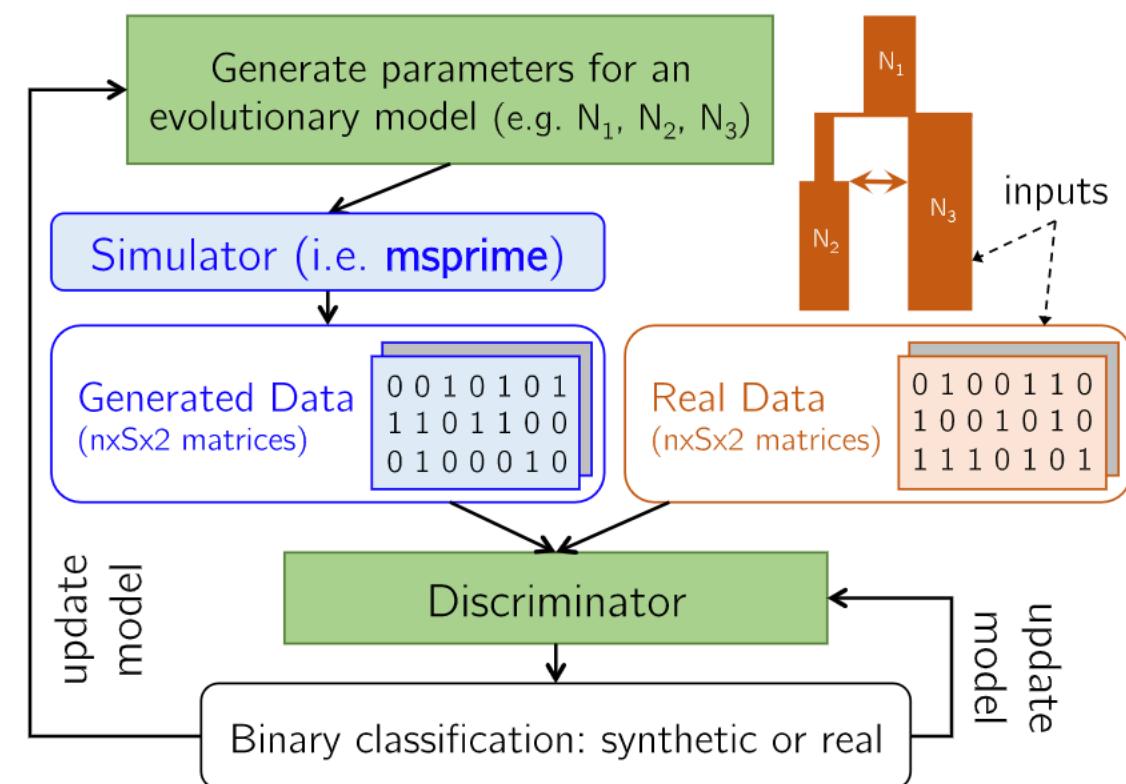
 Kevin Korfmann,  Thibaut Sellinger,  Fabian Freund,  Matteo Fumagalli,
 Aurélien Tellier



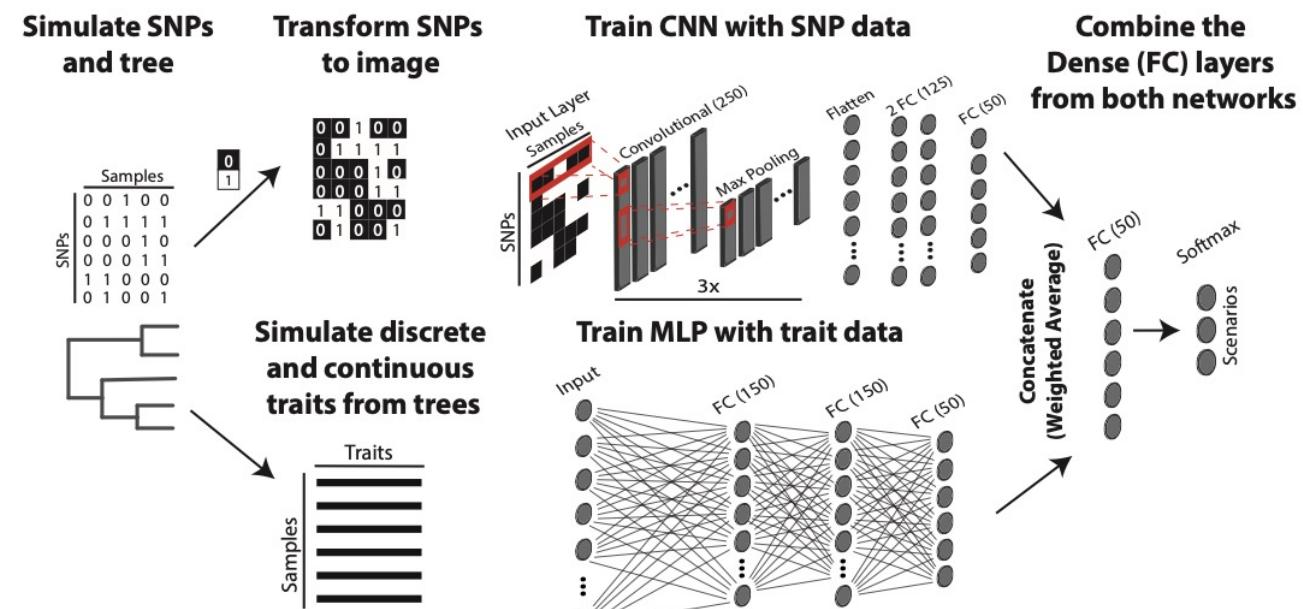
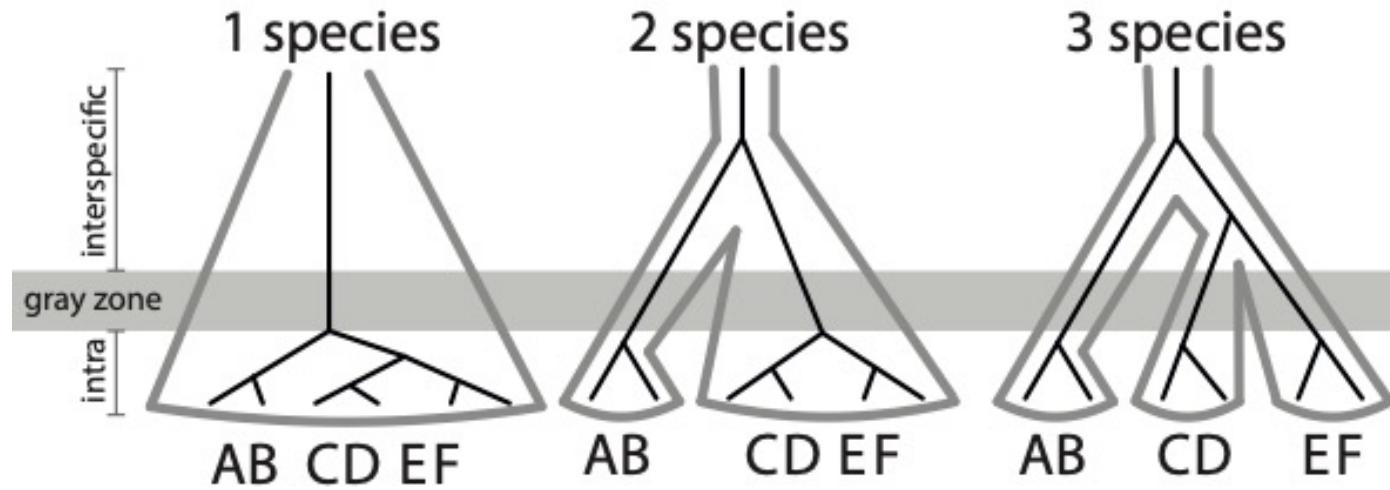
Hierarchical graph representation Ying et al. (2019) arXiv

Automatic inference of demographic parameters using generative adversarial networks

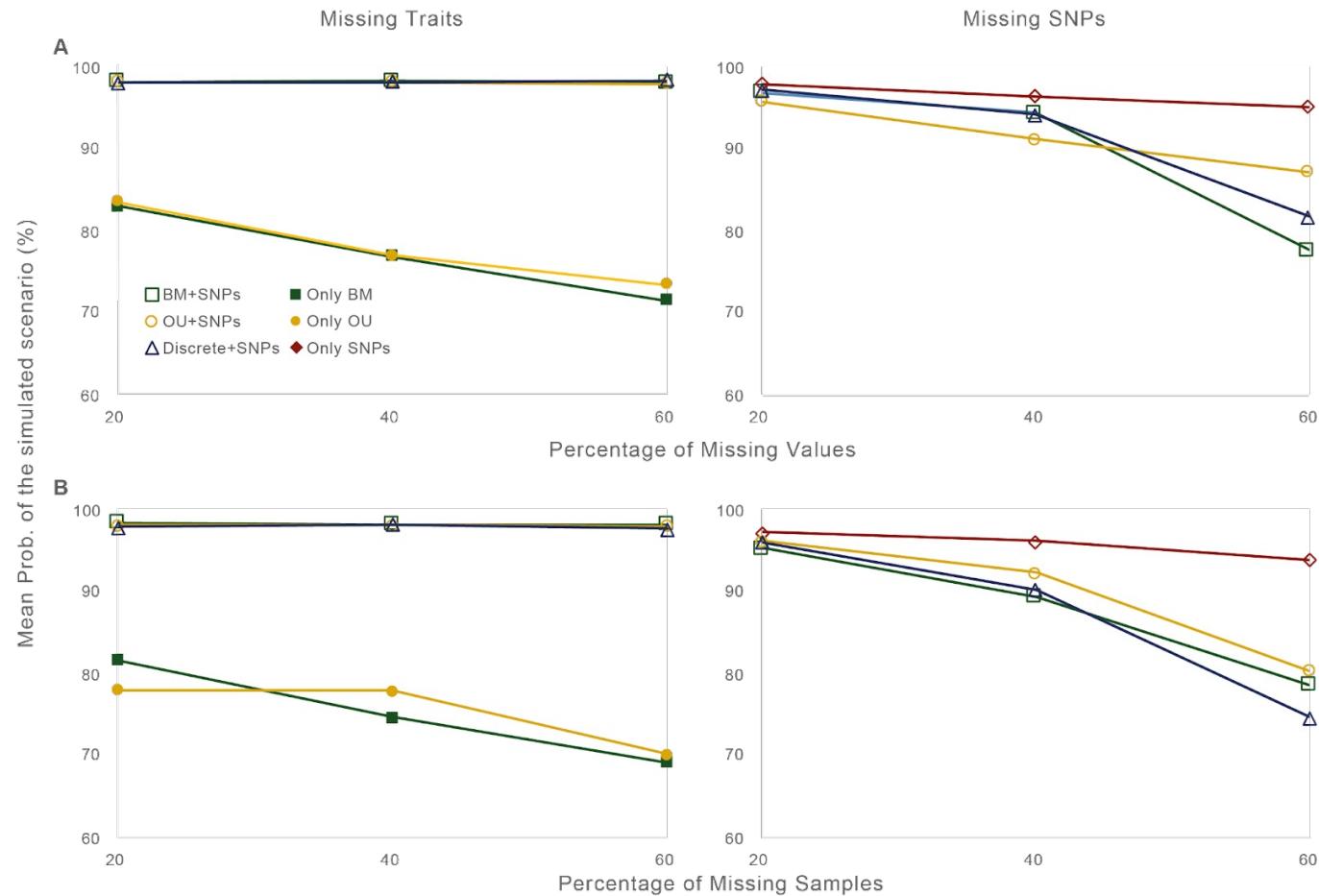
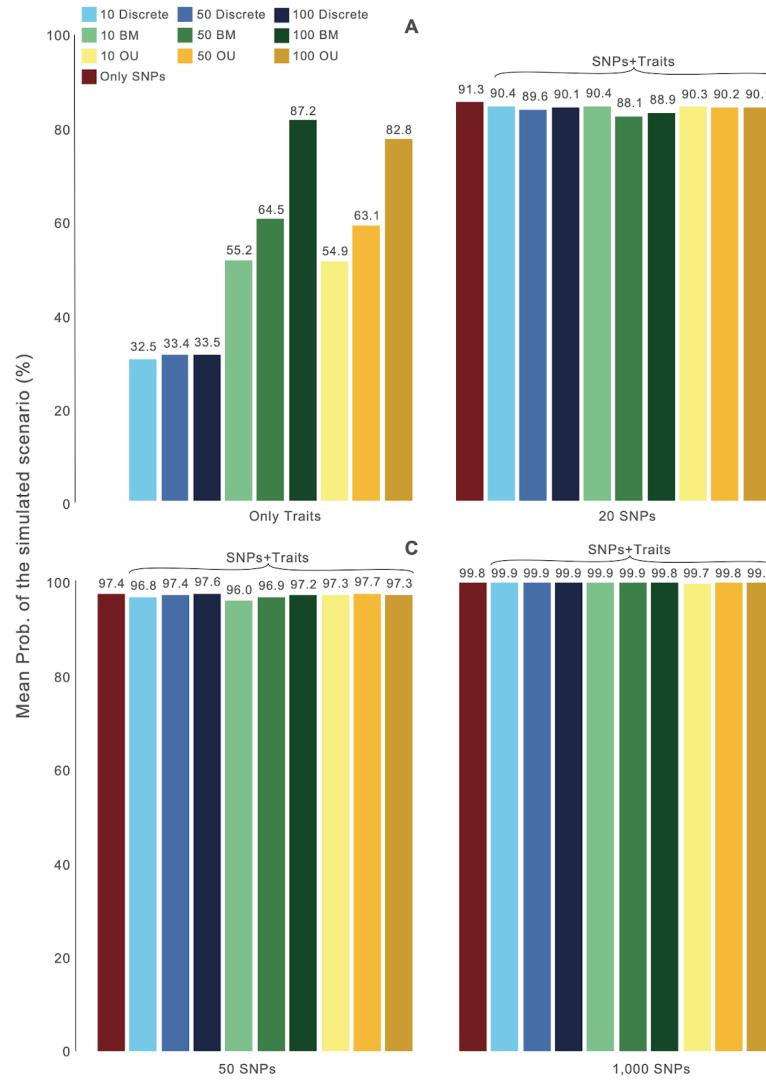
Zhanpeng Wang¹ | Jiaping Wang¹ | Michael Kourakos² | Nhung Hoang² |
Hyong Hark Lee² | Iain Mathieson³ | Sara Mathieson¹ 



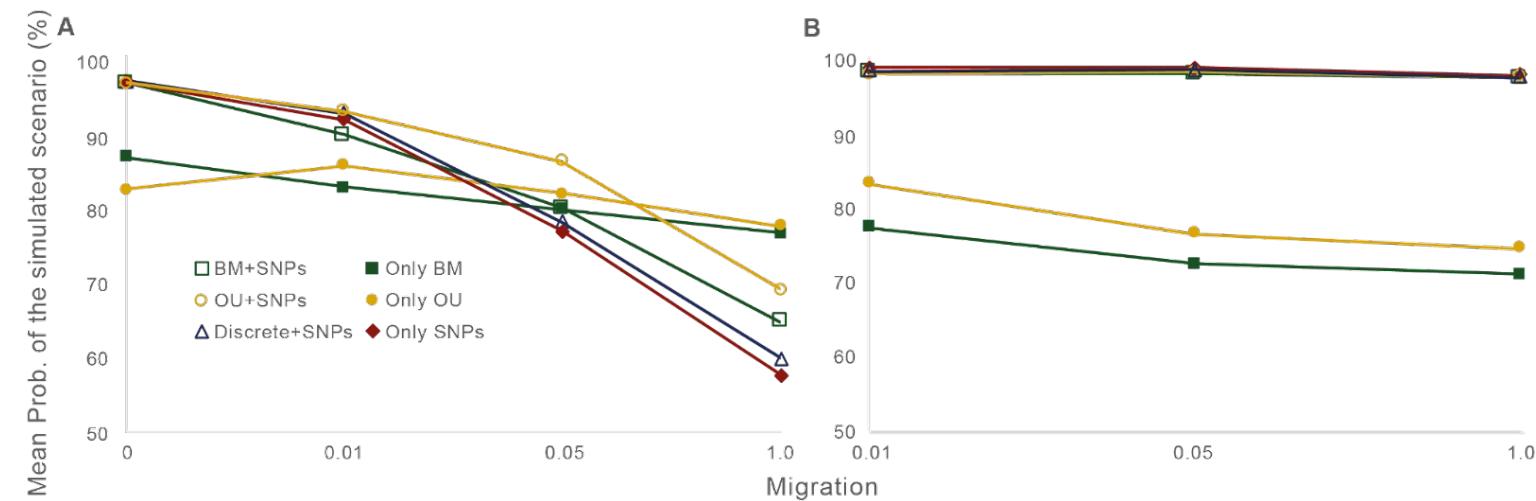
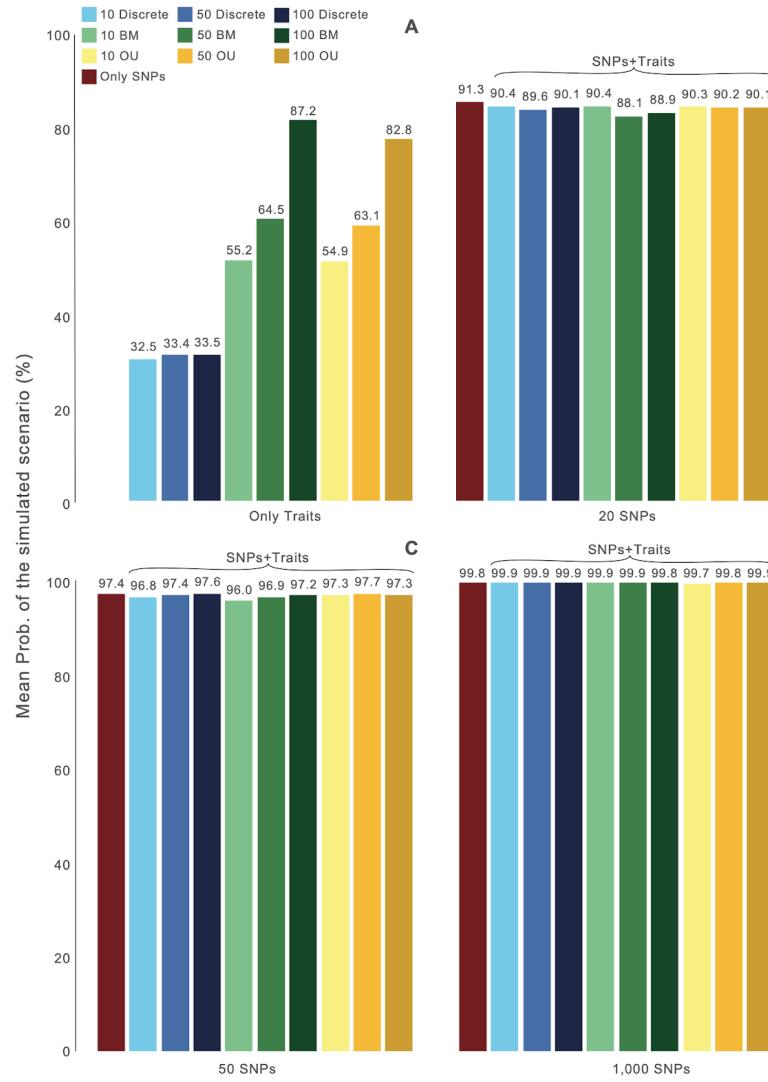
Integrative Deep Learning species delimitation



Integrative Deep Learning species delimitation

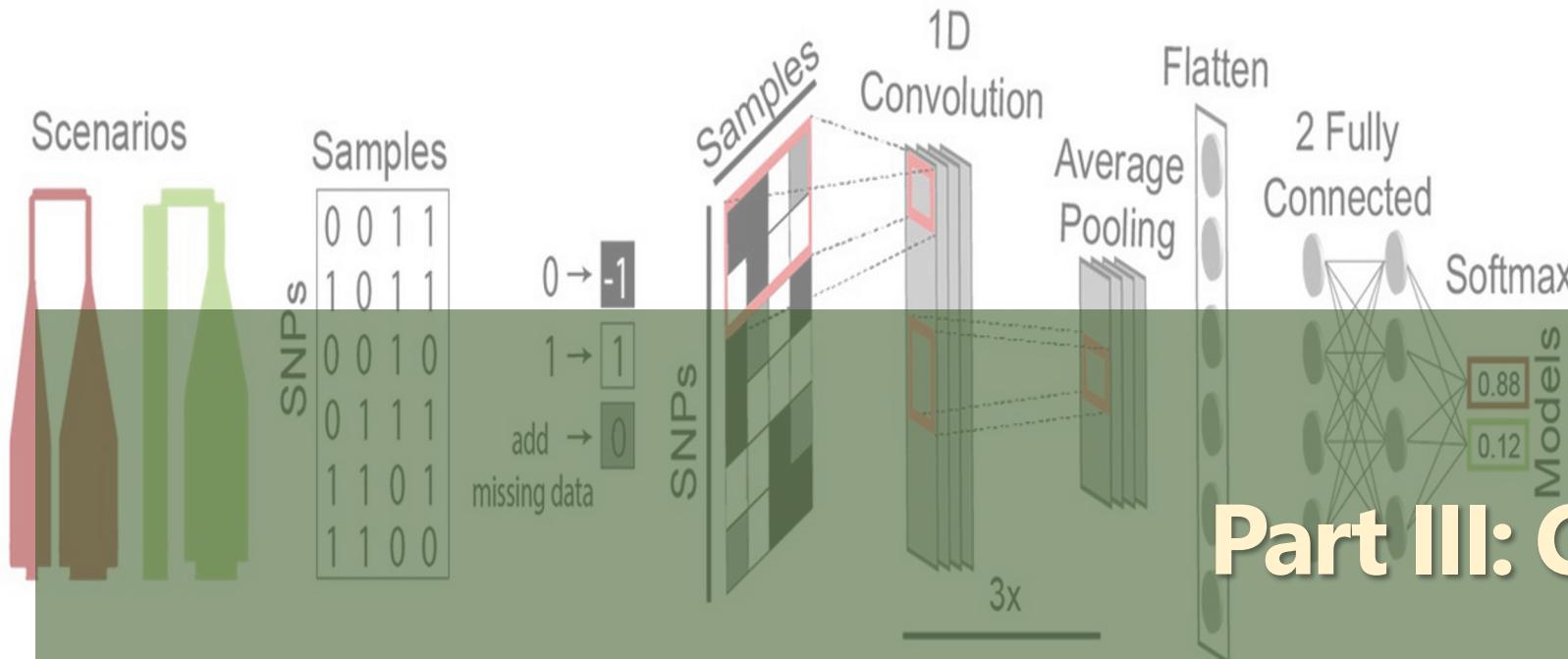


Integrative Deep Learning species delimitation



What's next?

Your
Project
Here



Part III: Course wrapup.

DEEP LEARNING METHODS IN POPULATION GENOMICS AND PHYLOGEOGRAPHY

Course goals

- Conceive and simulate genetic data under competing demographic scenarios 
- Understand deep learning background and how a CNN works 
- Simulate genomic regions with selective sweeps and use CNN to detect such regions on real genomes 
- How to use deep learning to compare demographic scenarios 

Thanks!



Deep Learning methods in population genomics and phylogeography

Manolo Fernandez Perez

 @ManoloLearning

 manolofperez@gmail.com

 sites.google.com/site/manolofperez

Matteo Fumagalli

 @MatteoFumagalli@fediscience.org

 m.fumagalli@qmul.ac.uk

 <https://mfumagalli.github.io/>