DEEP LEARNING FOR GENOMICS

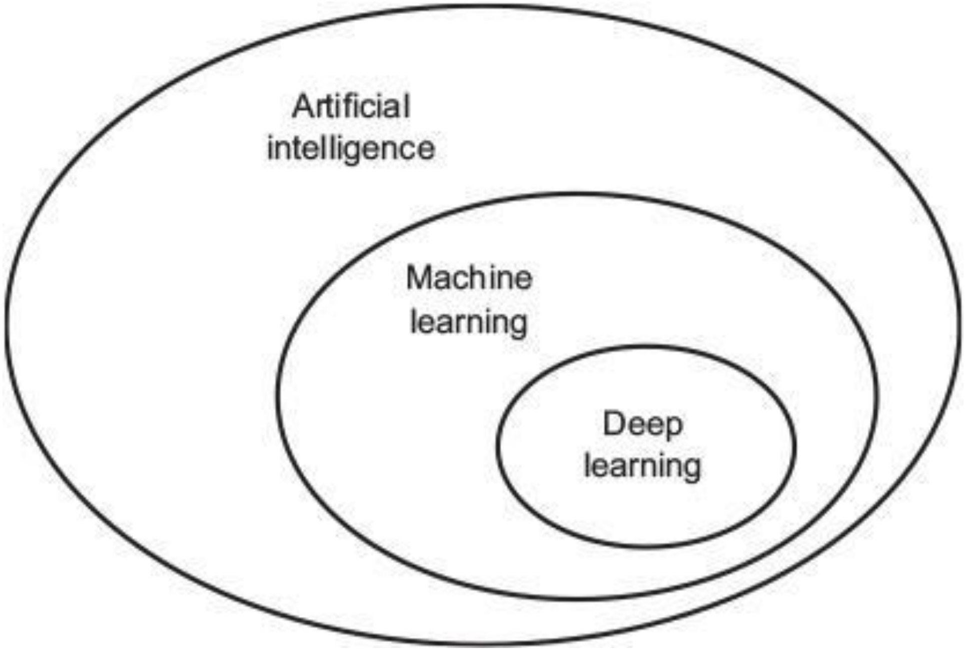
Raphaël MOURAD, Assistant Professor,
Visiting professor at MIAT Toulouse (MathNum)
University Paul Sabatier, Toulouse III

WHAT IS DEEP LEARNING?

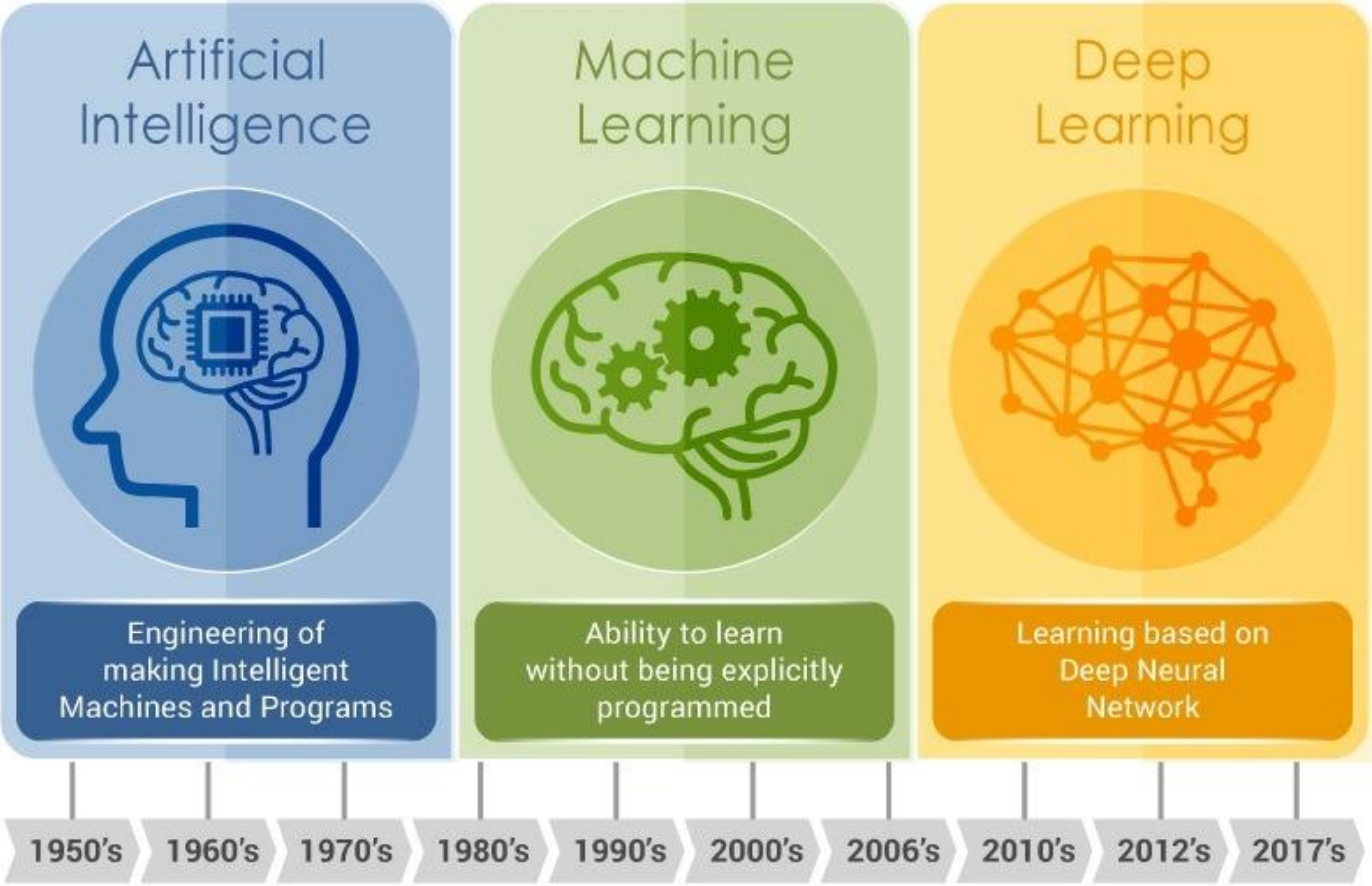
Deep learning as a branch of AI

- Deep learning is a branch of machine learning and AI, which has been very successful in the past years (since 2012).

- Deep learning relies on:
  - new algorithms,
  - new GPUs
  - access to big data.

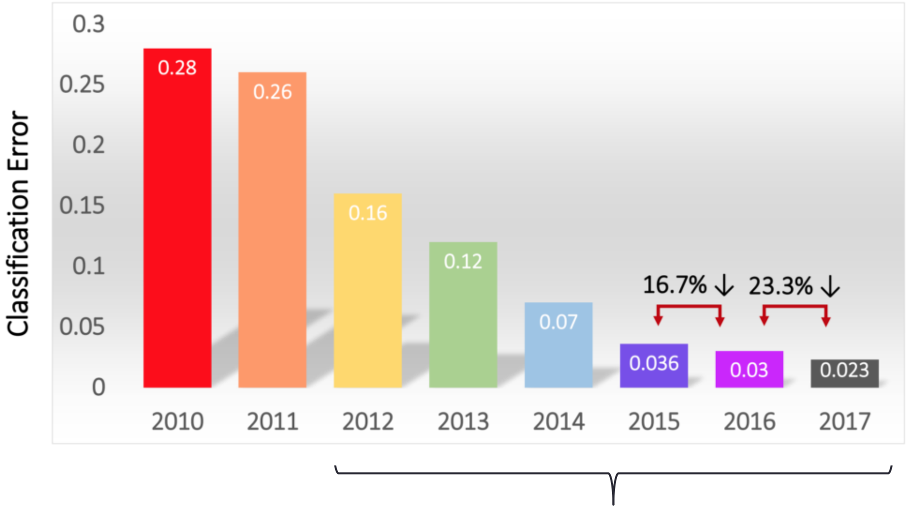


Deep learning as a branch of AI


Artificial Intelligence
Machine Learning
Deep Learning
Engineering of making Intelligent Machines and Programs
Ability to learn without being explicitly programmed
Learning based on Deep Neural Network
1950's
1960's
1970's
1980's
1990's
2000's
2006's
2010's
2012's
2017's

Success of deep learning since 2012: Example of computer vision

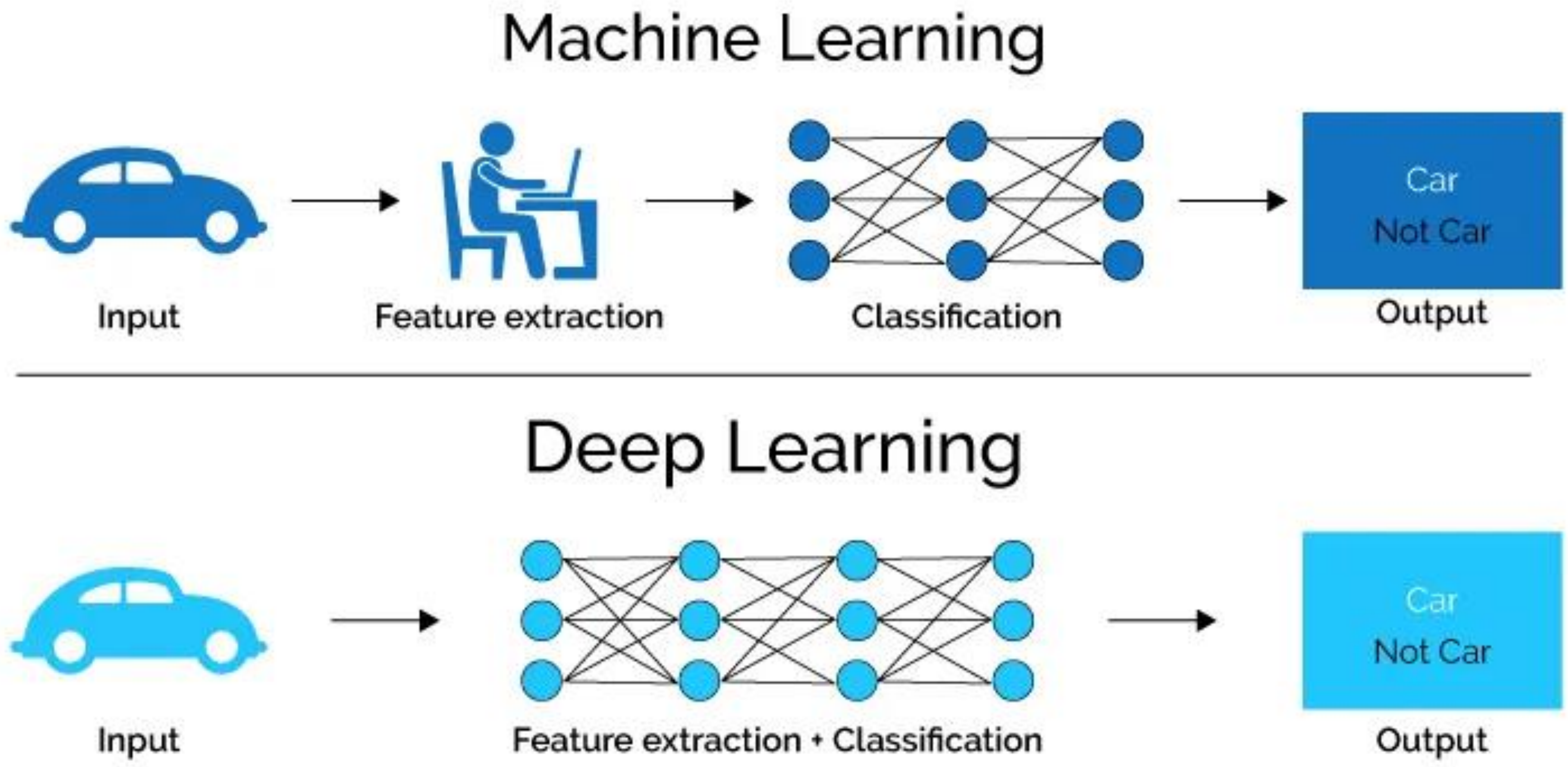
Image classification
(ImageNet challenge)



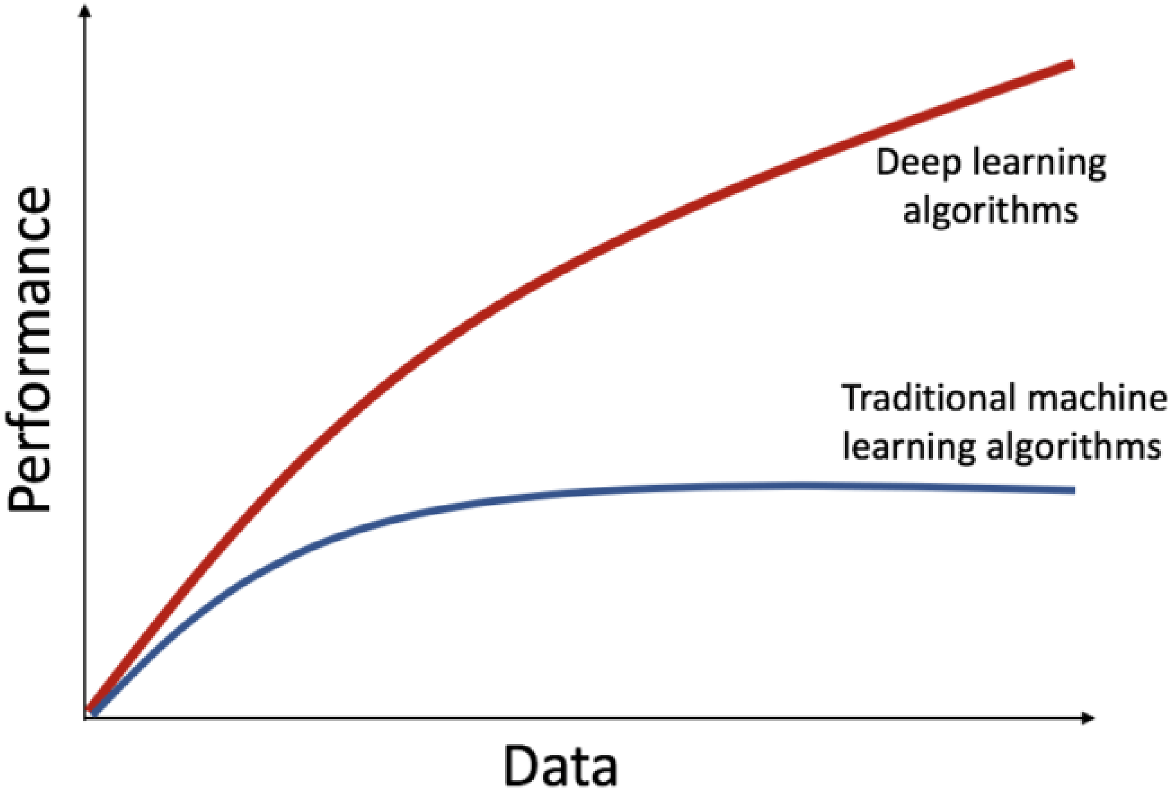
With deep learning

- 2012: AlexNet (convNet)
- 2013: ZFNet
- 2014:
  - VGGNet (deeper, simpler)
  - InceptionNet (faster)
- 2015: ResNet (deeper)
- 2016: Ensemble networks

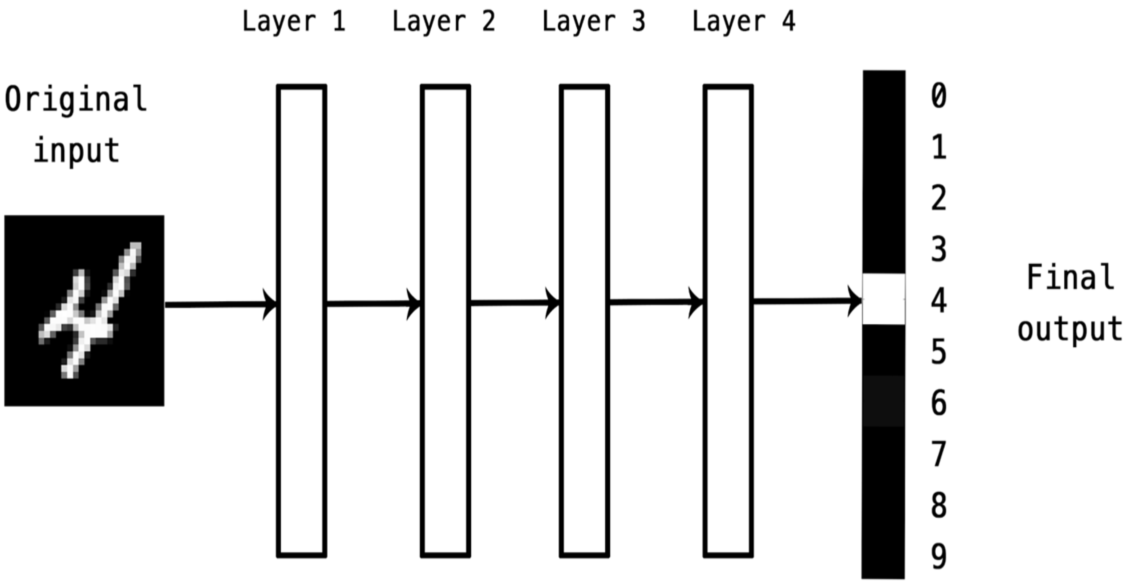
Difference between machine and deep learning



Difference between machine and deep learning

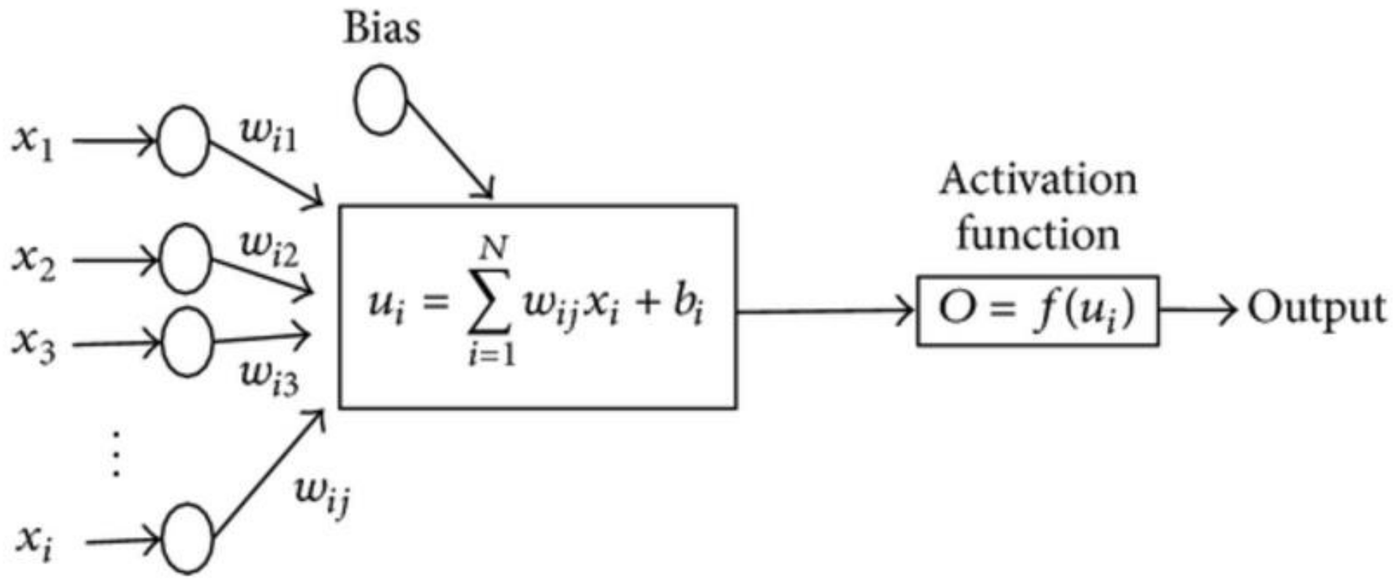


Deep learning as neural networks



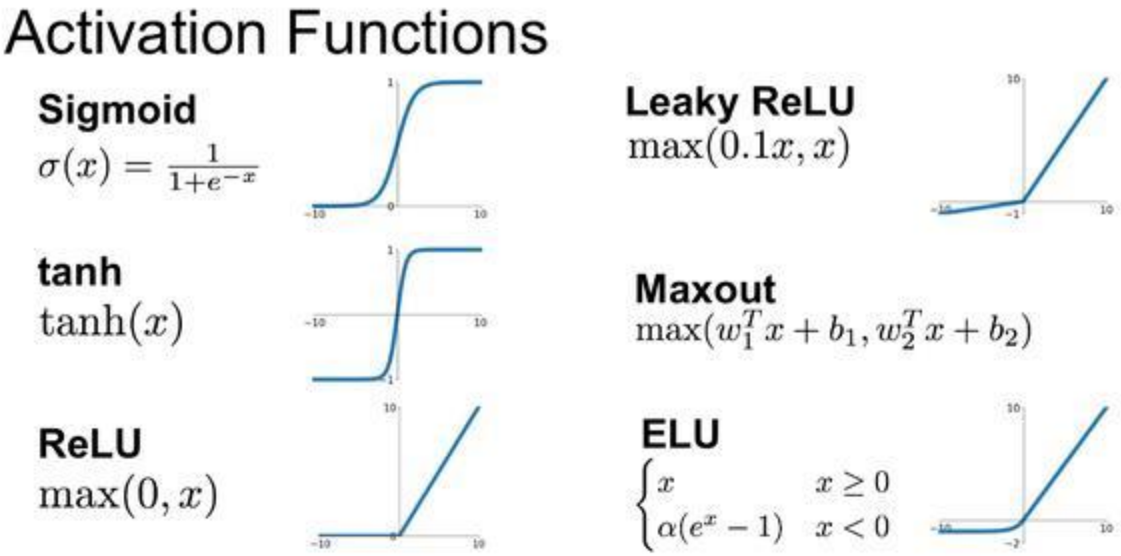
- Deep learning is based on a **deep** neural network which is the stacking of different neuronal layers to predict a final output.

Neural networks (not deep)



- In a neural network, multiple inputs x_i are combined through a linear combination (with weights w_i), and then an activation function is used for a non-linear transformation to obtain the output.

Activation function



Activation Functions

Sigmoid
$\sigma(x) = \frac{1}{1+e^{-x}}$

tanh
$\tanh(x)$

ReLU
$\max(0, x)$

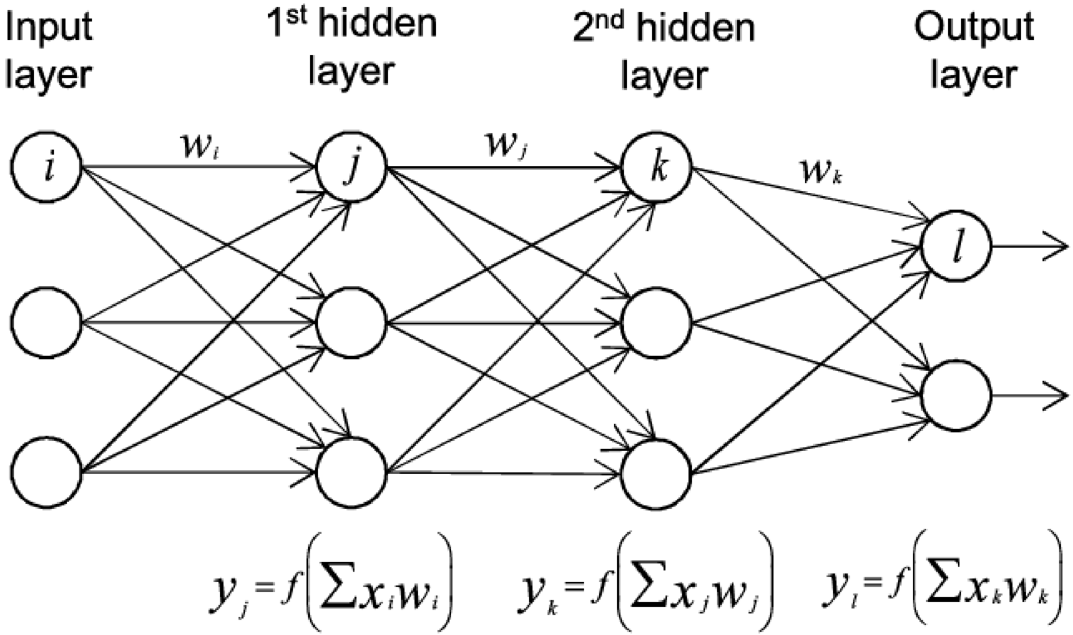
Leaky ReLU
$\max(0.1x, x)$

Maxout
$\max(w_1^T x + b_1, w_2^T x + b_2)$

ELU
$\begin{cases} x & x \geq 0 \\ \alpha(e^x - 1) & x < 0 \end{cases}$

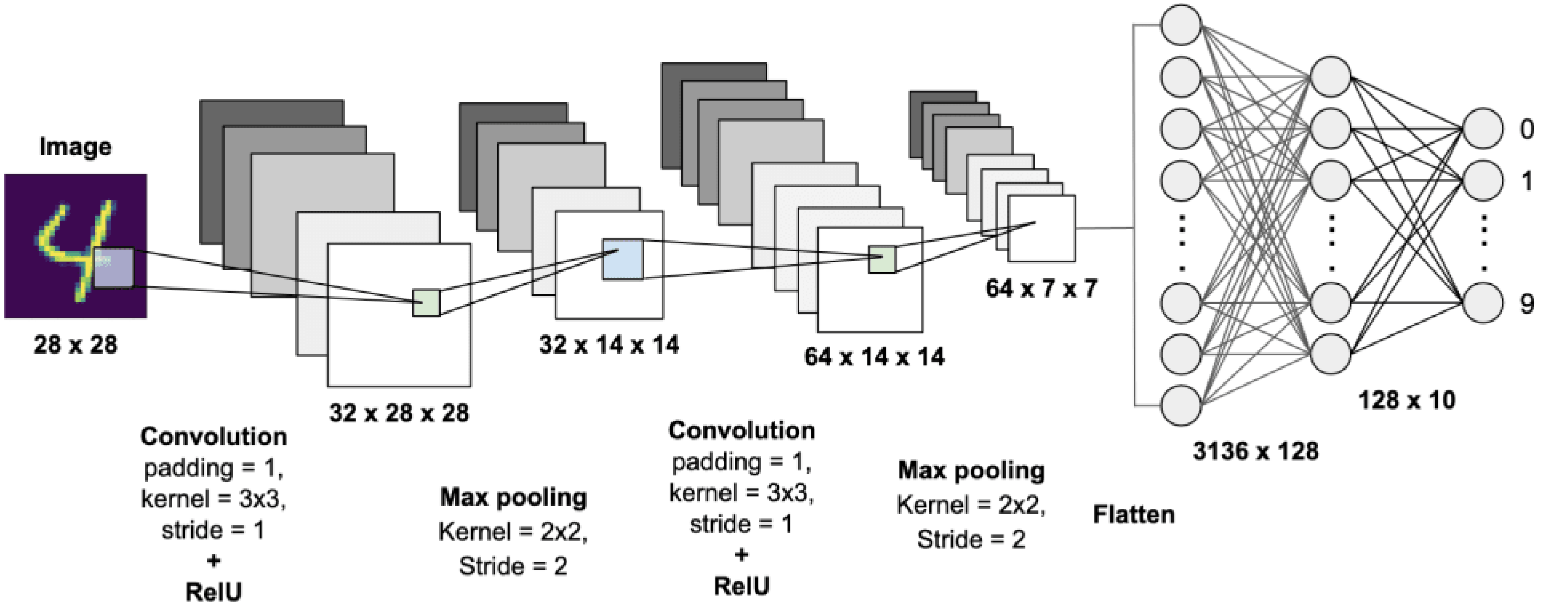
- The activation function allows to obtain a non-linear output from a linear input.
- NB: the linear activation function also exists (A = cx).

Deep neural networks


Input layer
1st hidden layer
2nd hidden layer
Output layer
w_i
w_j
w_k
i
j
k
l
$y_j = f\left(\sum x_i w_i\right)$
$y_k = f\left(\sum x_j w_j\right)$
$y_l = f\left(\sum x_k w_k\right)$

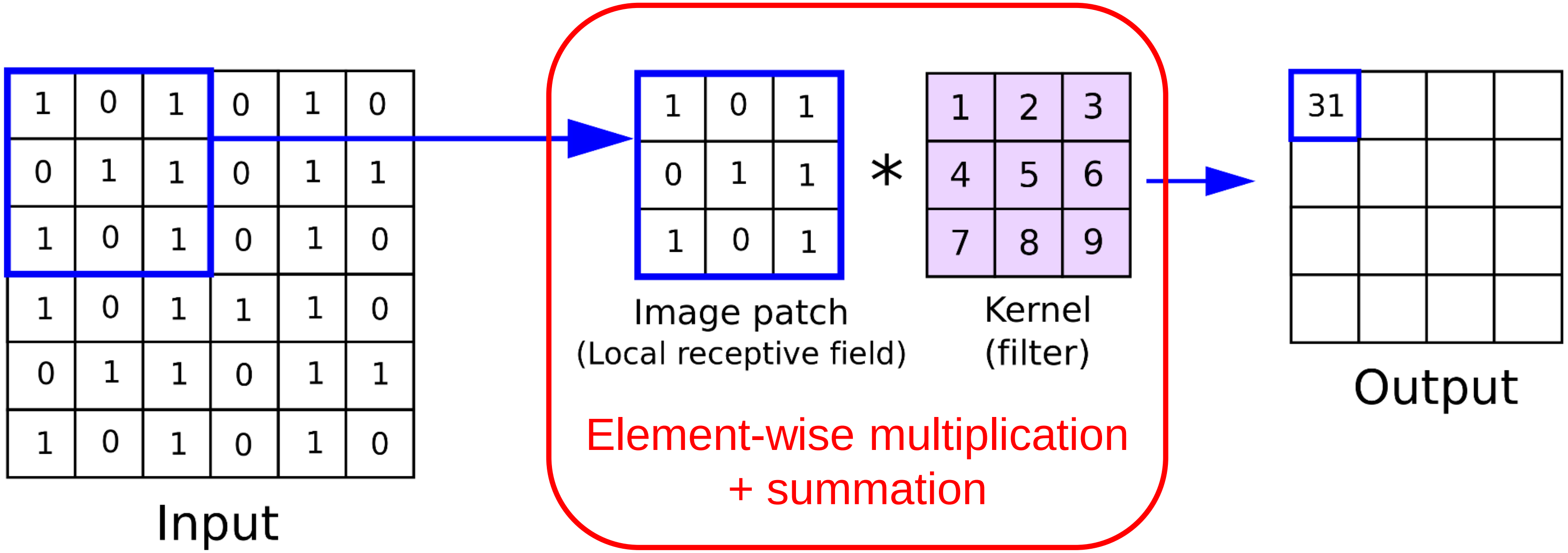
- A deep neural network (DNN) is a neural network (NN) with multiple layers between the input and output layers. Each hidden layer linearly combines the output from the previous layer and then does a non-linear transformation.

Convolutional neural networks (CNNs)



- A CNN is based on the stacking of one or more convolutional layers, followed by one or more dense layers (dense layer = classical neural network layer).

Convolutional layer



1	0	1	0	1	0
0	1	1	0	1	1
1	0	1	0	1	0
1	0	1	1	1	0
0	1	1	0	1	1
1	0	1	0	1	0

Input

1	0	1
0	1	1
1	0	1

Image patch
(Local receptive field)

*

1	2	3
4	5	6
7	8	9

Kernel
(filter)

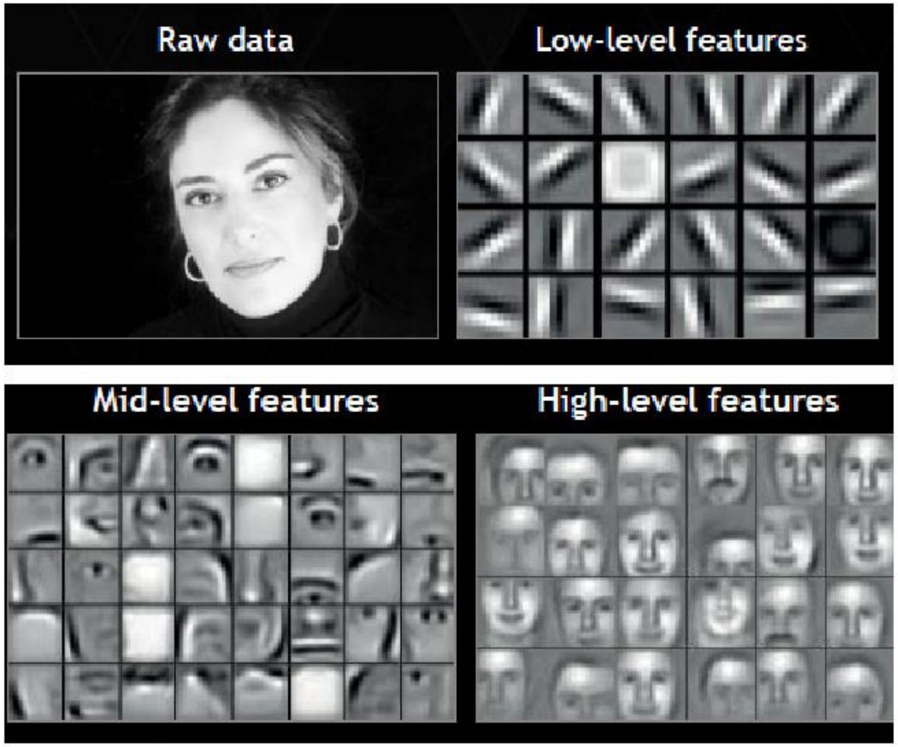
Element-wise multiplication
+ summation

31			

Output

- A patch (submatrix) in the input matrix is multiplied by a kernel (or filter) to obtain an output value. This operation is done for every patch to obtain every output value.

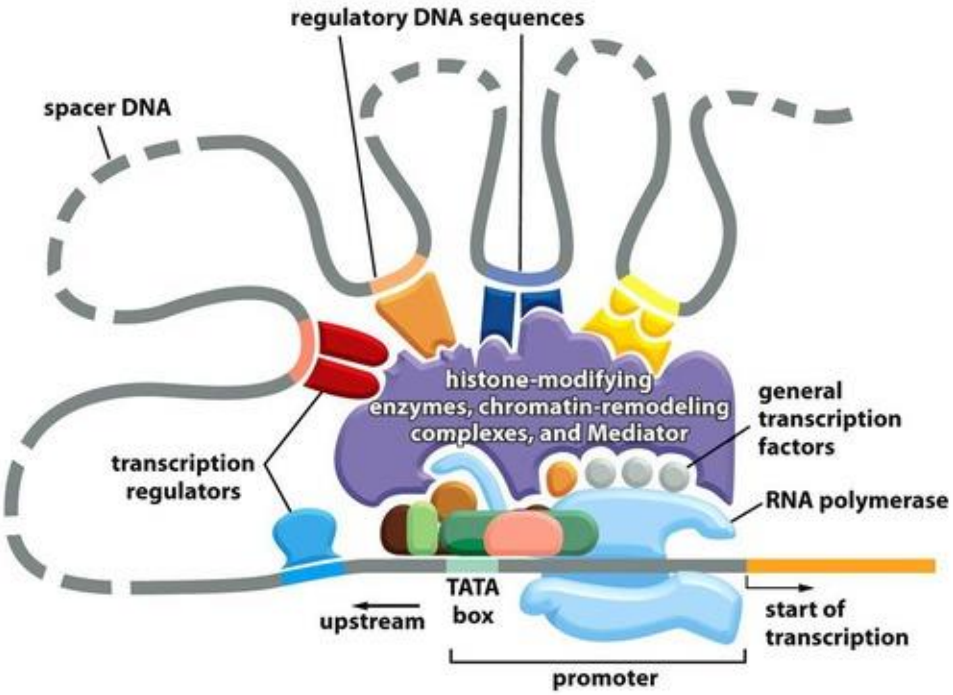
What do the kernels represent in the convolutional layer(s)?



- In the first conv layer, the kernels correspond to low-level features (often edges). In the middle conv layers, the kernels correspond to mid-level features (parts of an object). In the last conv layers, the kernels correspond to high-level features (often objects).

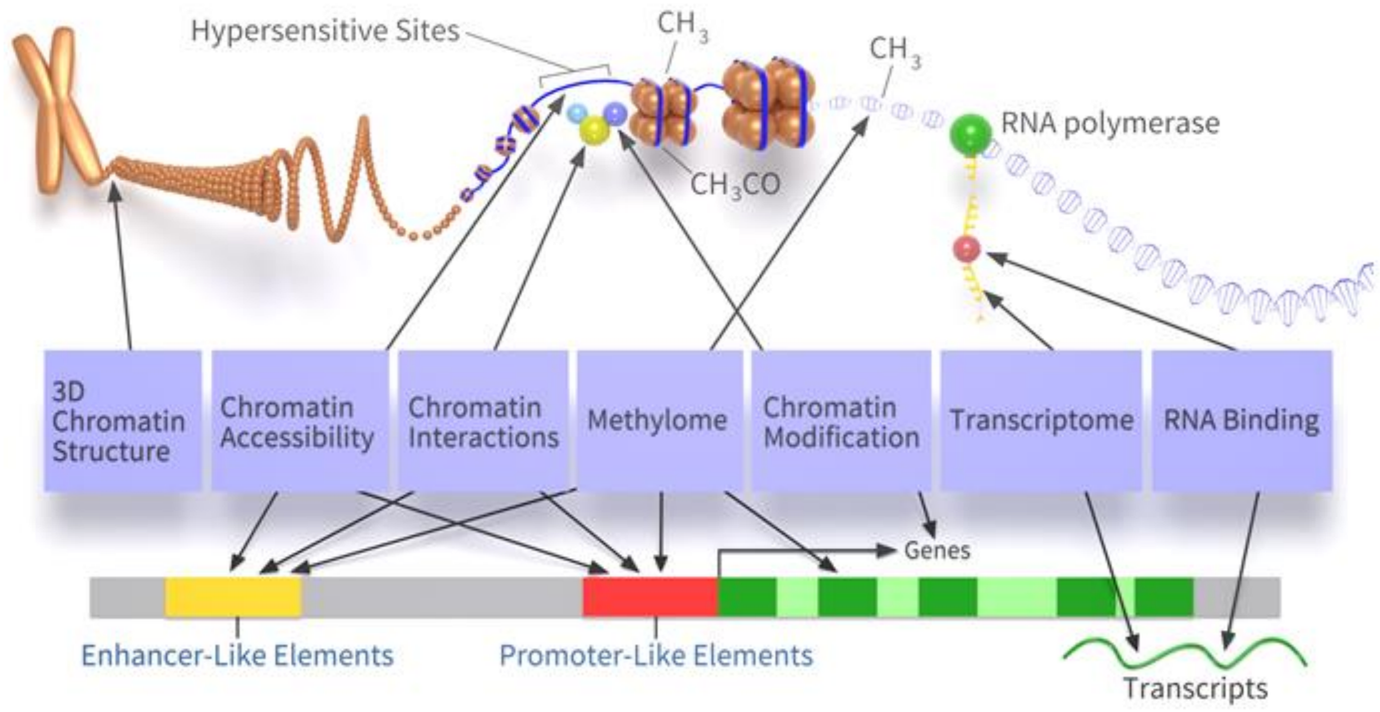
REGULATORY SEQUENCE

Regulatory regions



- Regulatory regions (promoters, enhancers, insulators, …) are non-coding DNA sequences that control the expression of target genes.

Regulatory regions

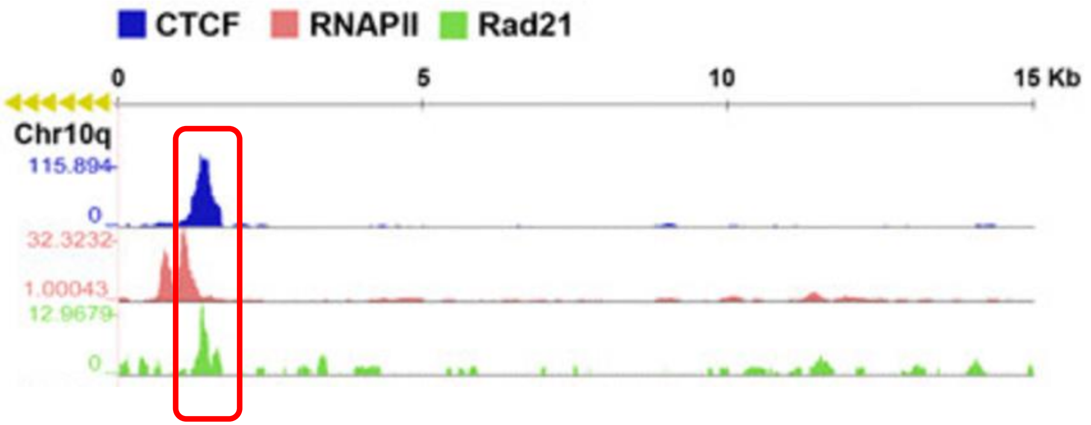


- Regulatory regions were mapped during the last decade using techniques such as ChIP-seq, ATAC-seq, Hi-C, methyl-seq...

Regulatory elements regulate many other processes

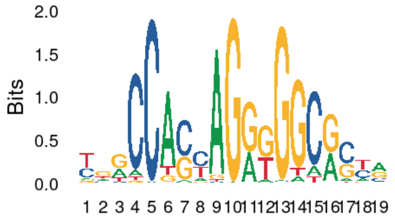
- Regulatory elements regulate:
  - Gene expression
  - DNA replication (origins of replication)
  - DNA recombination (recombination hotspots)
  - Heterochromatin formation and dynamics (polycomb,…)
  - 3D chromatin structure (CTCF-mediated looping
  - …

CTCF ChIP-seq peaks as examples



CTCF peak

- We extract the sequences of the CTCF ChIP-seq peaks.
- If we run a motif search (using MEME for instance), we will observe the CTCF motif MA0139.1:



DEEP LEARNING FOR GENOMICS

CNN for classifying DNA sequences

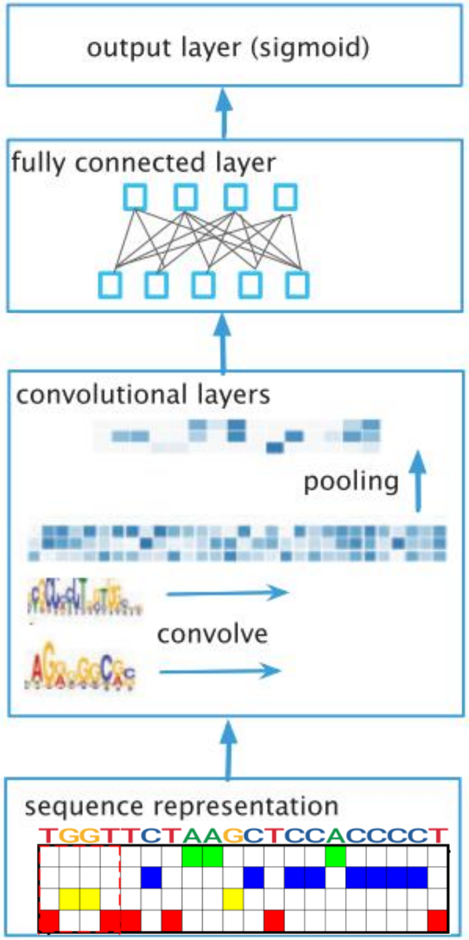
- Binary output

- Standard fully connected layer

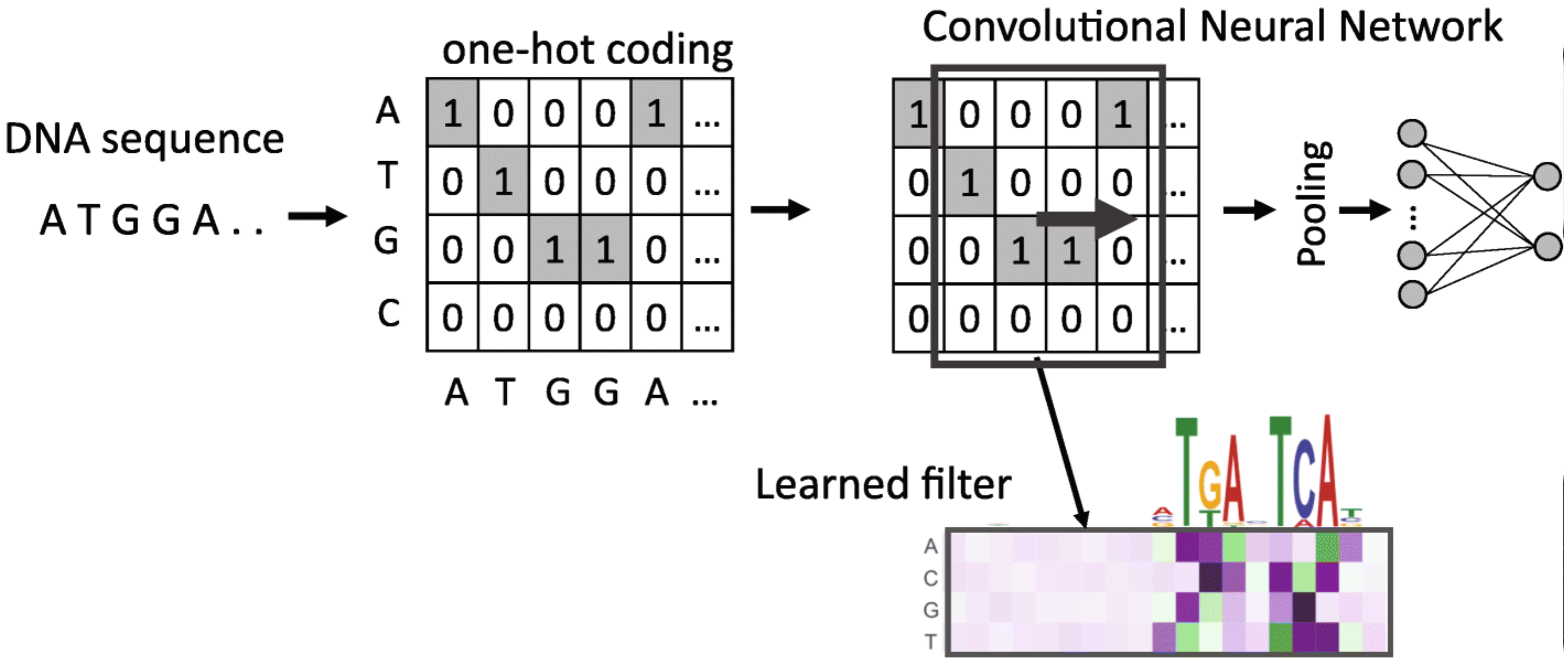
- 1-dimension convolution

  = DNA motif scanning

- One-hot encoding of DNA:
  Colored cells = 1; white cells = 0.



Meaning of 1D-convolution for DNA sequences

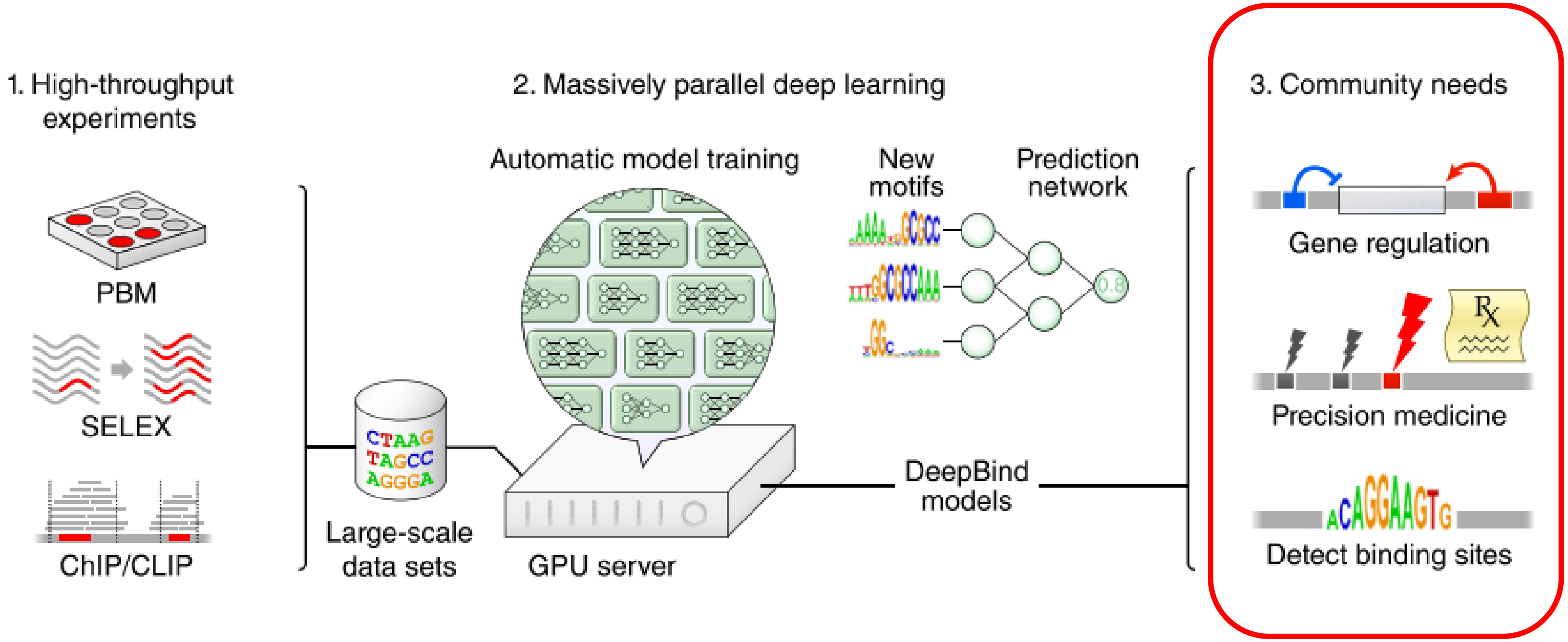


	A	T	G	G	A	...
A	1	0	0	0	1	...
T	0	1	0	0	0	...
G	0	0	1	1	0	...
C	0	0	0	0	0	...

- Based on kernels (filters) that are matrices of weights for each base of a DNA motif.
- Kernels = Position Weight Matrices (given some transformation).

Example: Deepbind

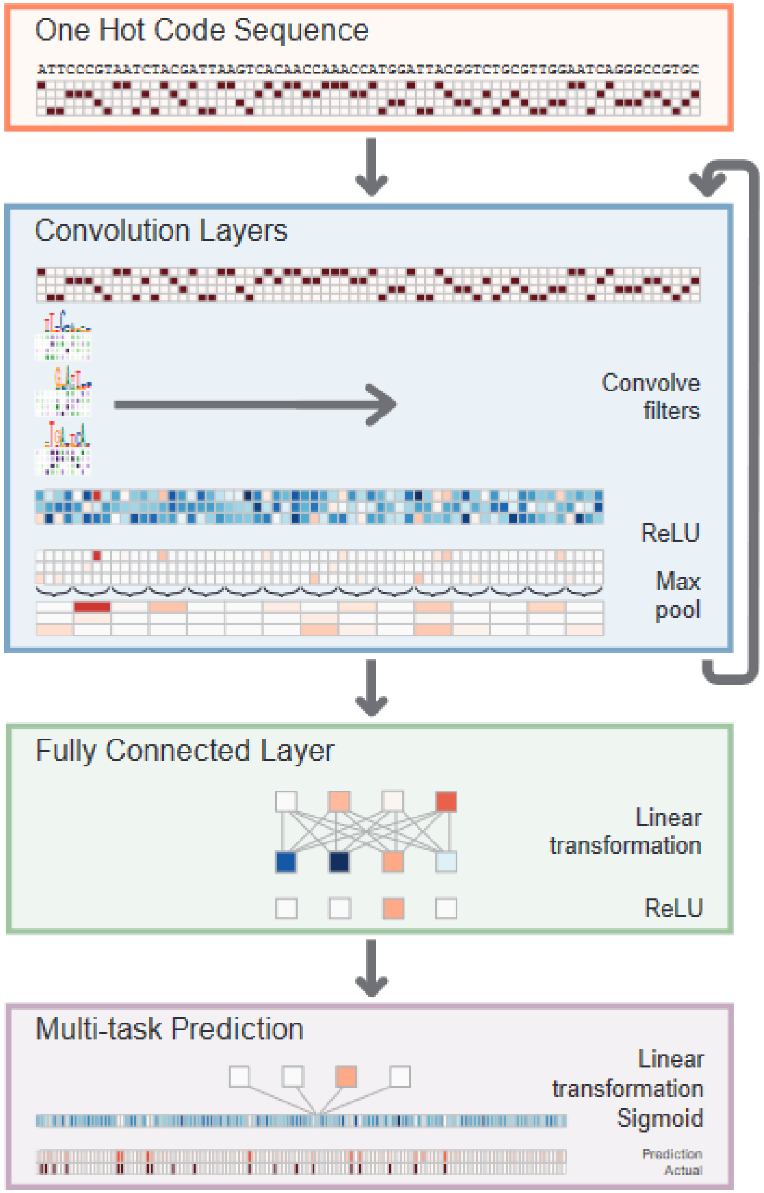
Applications in biology and personalized medicine



- Predict binding proteins to DNA given the DNA sequence.

Basset

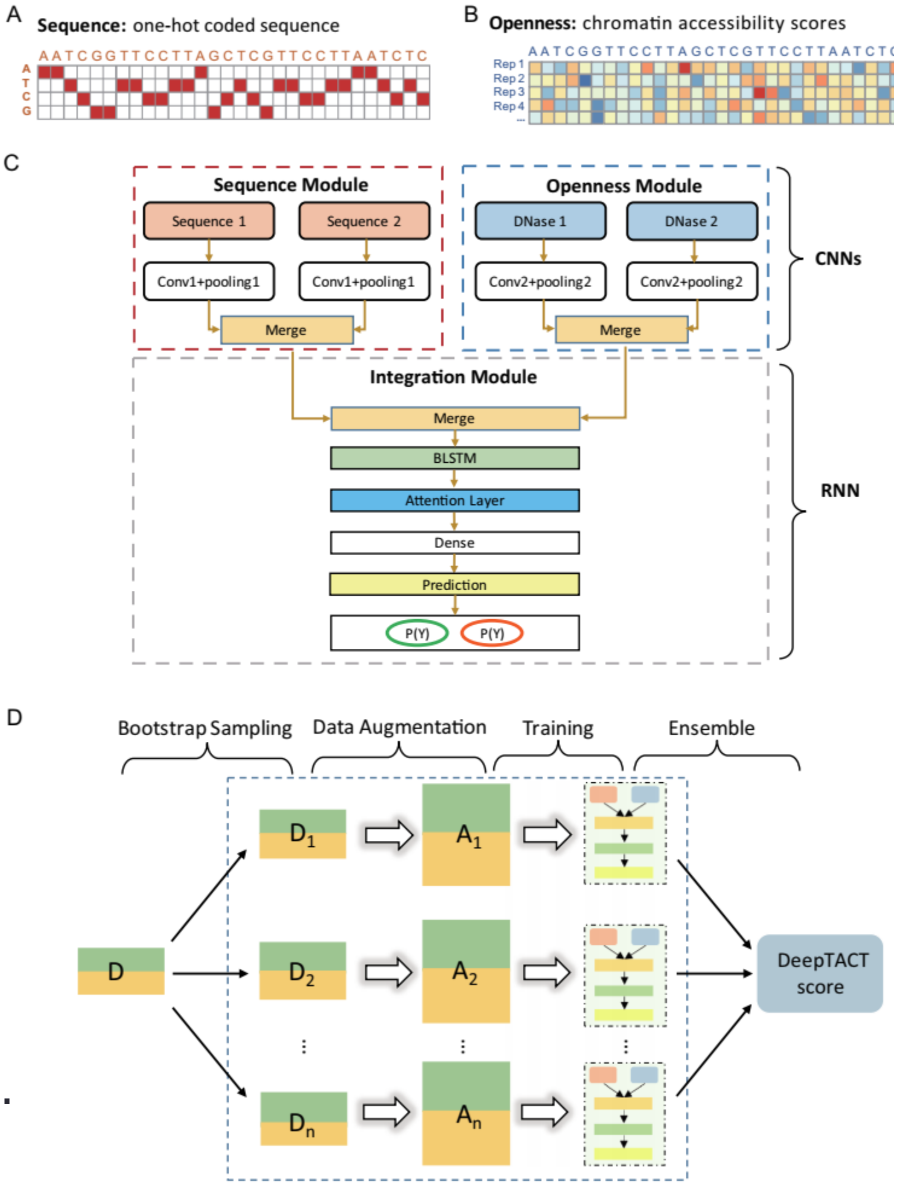
- Predict chromatin accessibility (DNase-seq) from DNA sequences.



Kelley et al. Genome Res 2016.

DeepTact

- Predict long-range contacts (Hi-C) from DNA sequences and chromatin accessibility.

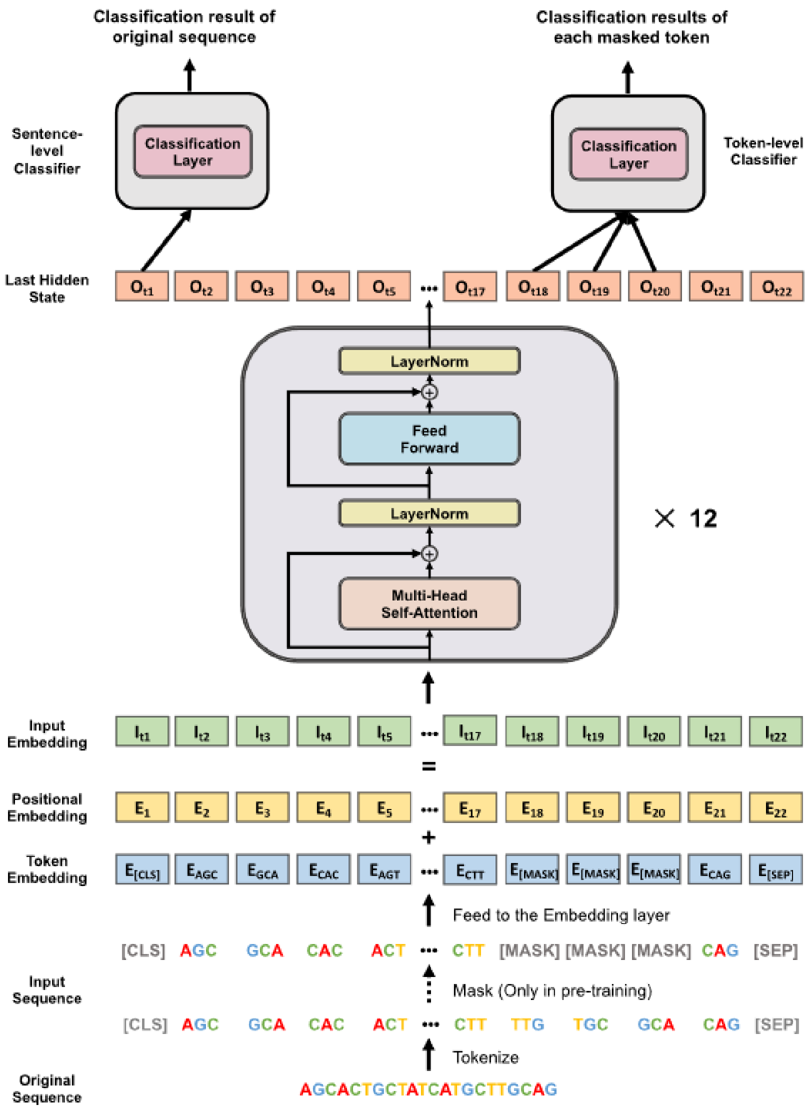


Li et al. Nucleic Acids Research, 2019.

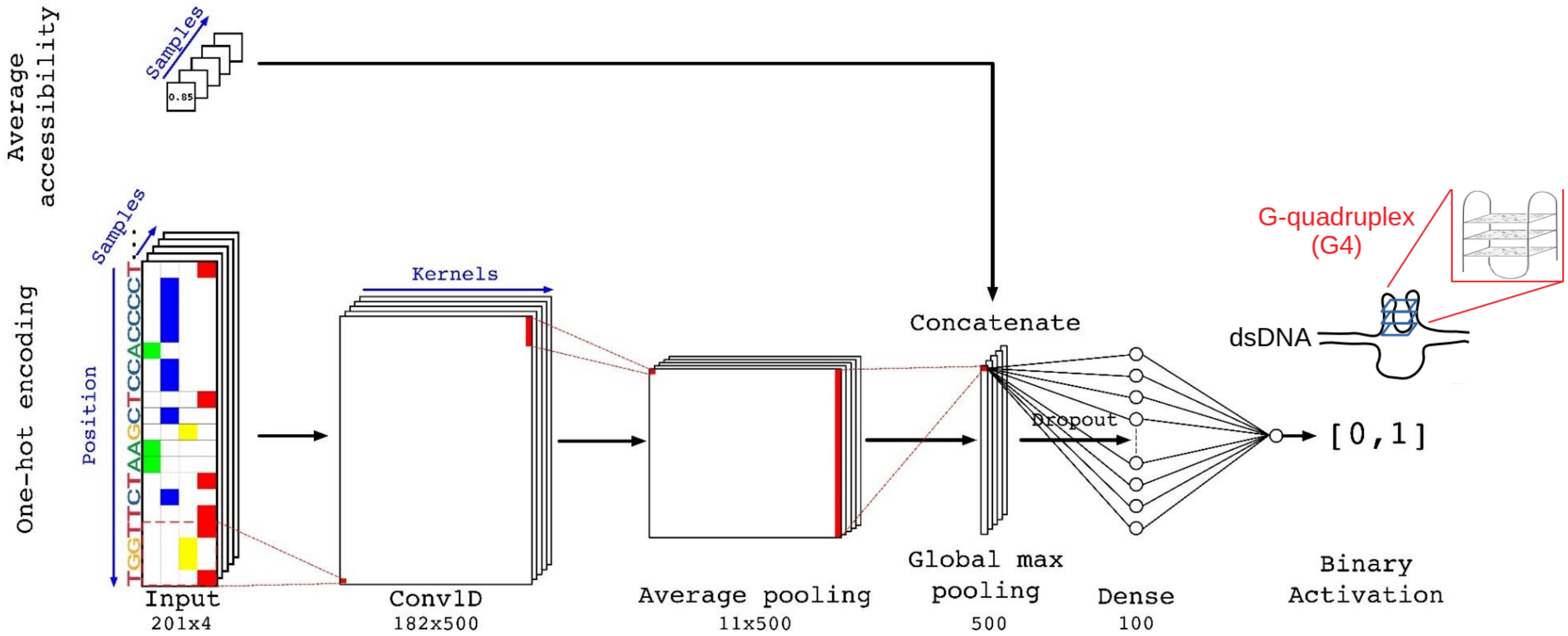
DNABERT

- The self-attention model DNABERT is trained by masking some kmers in the DNA sequence and then by trying to predict them using the other k-mers in the DNA sequence (context).
- At the end, the model provides features that encode DNA sequences in a very efficient way for any predictive task.

Ji et al. Bioinformatics 2021.


Classification result of original sequence
Classification results of each masked token
Sentence-level Classifier
Classification Layer
Classification Layer
Token-level Classifier
Last Hidden State
O_t1 O_t2 O_t3 O_t4 O_t5 ··· O_t17 O_t18 O_t19 O_t20 O_t21 O_t22
LayerNorm
Feed Forward
LayerNorm
Multi-Head Self-Attention
× 12
Input Embedding
I_t1 I_t2 I_t3 I_t4 I_t5 ··· I_t17 I_t18 I_t19 I_t20 I_t21 I_t22
=
Positional Embedding
E_1 E_2 E_3 E_4 E_5 ··· E_17 E_18 E_19 E_20 E_21 E_22
+
Token Embedding
E_[CLS] E_AGC E_GCA E_CAC E_AGT ··· E_CTT E_[MASK] E_[MASK] E_[MASK] E_CAG E_[SEP]
Feed to the Embedding layer
[CLS] AGC GCA CAC ACT ··· CTT [MASK] [MASK] [MASK] CAG [SEP]
Input Sequence
Mask (Only in pre-training)
[CLS] AGC GCA CAC ACT ··· CTT TTG TGC GCA CAG [SEP]
Tokenize
Original Sequence
AGCACTGCTATCATGCTTGCAG

DeepG4

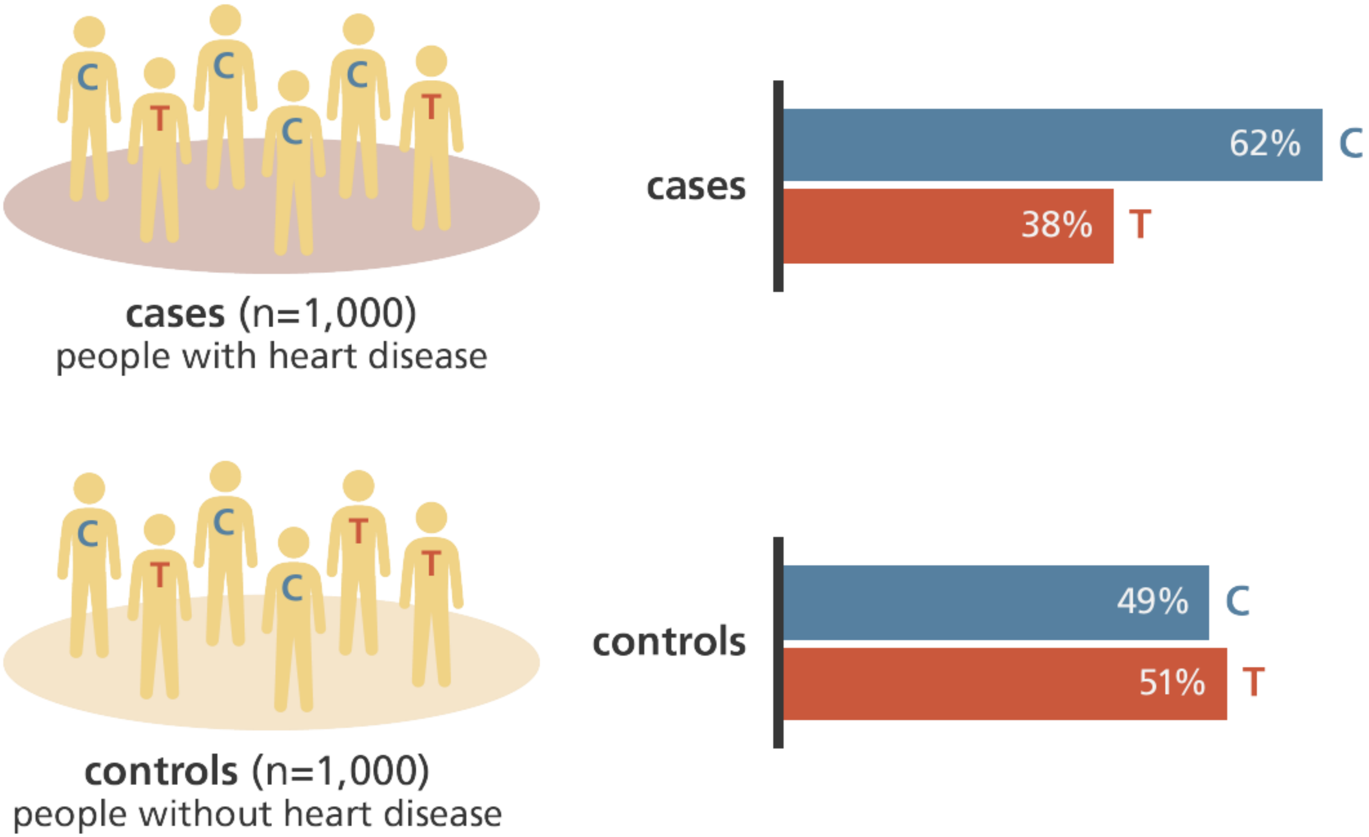


- Predict cell-type specific G-quadruplex structures given the DNA sequence and chromatin accessibility.

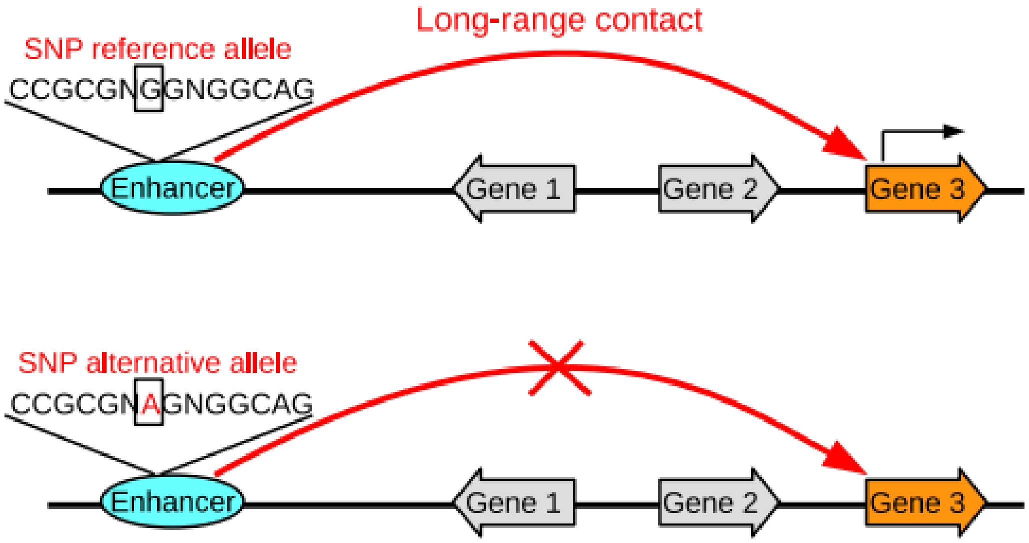
Rocher, Genais, Nassereddine and Mourad. PLOS Comp Bio 2021.

PREDICTION OF THE IMPACT OF MUTATIONS

Genome-wide association studies and SNPs

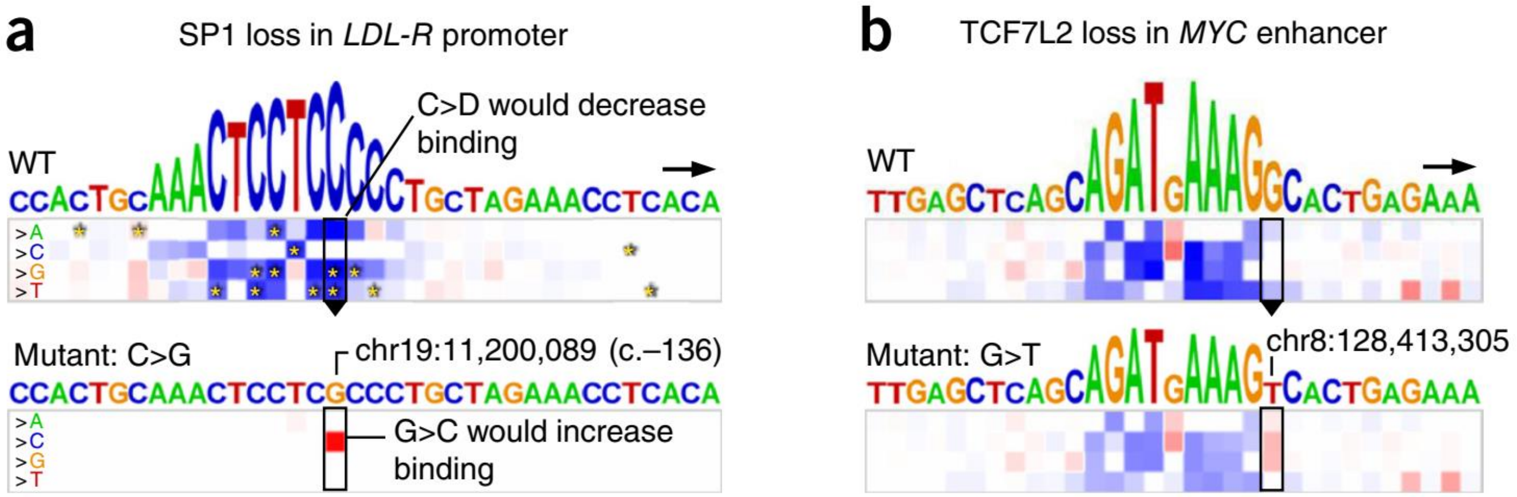


Regulatory elements are involved in genetic diseases (GWASs)



- > 95% of associated SNPs are located outside coding sequences.
- 75% of these SNPs overlap DNase I hypersensitive sites, which suggests their association with regulatory elements.

The impact of SNPs can be assessed using deep CNNs

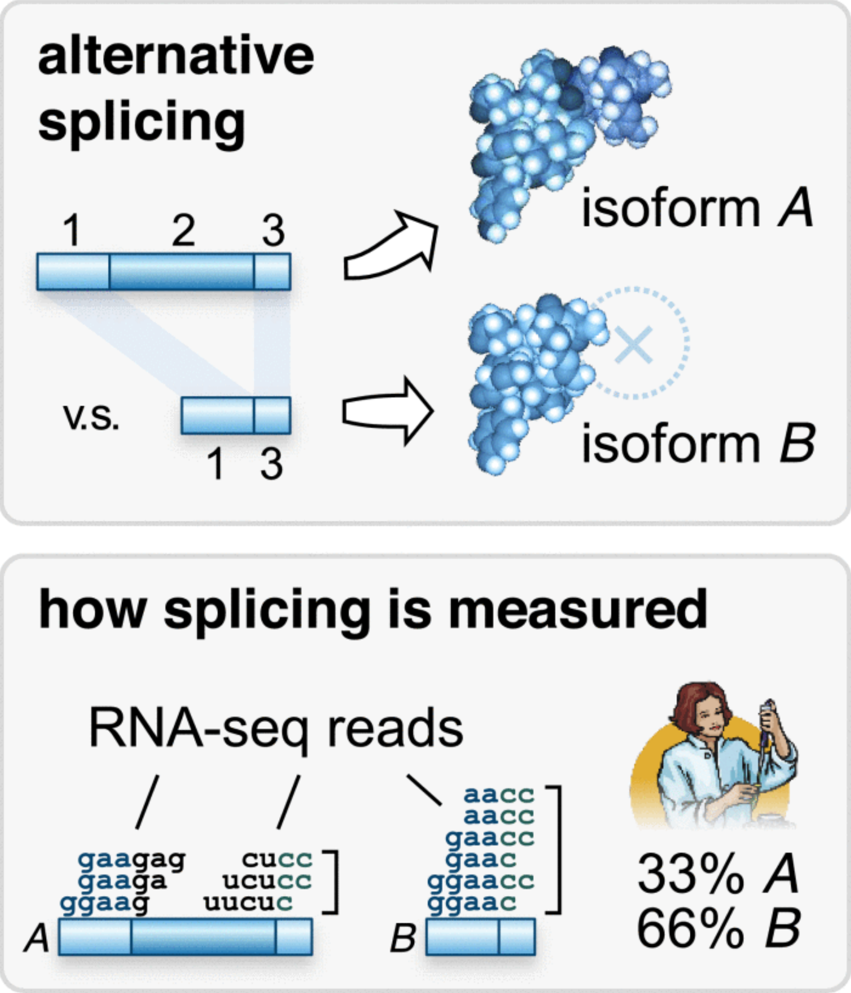


SNP associated to familial hypercholesterolemia.

A cancer risk SNP in a *MYC* enhancer.

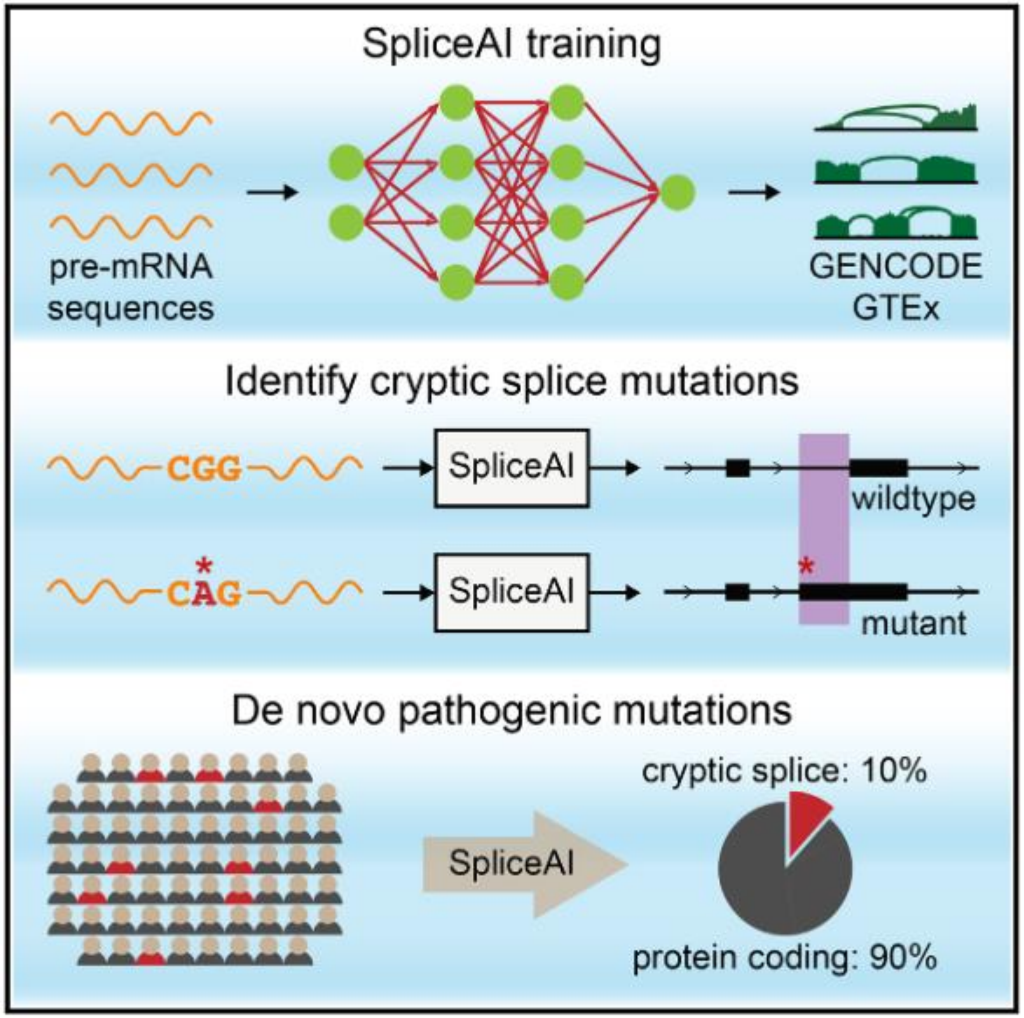
- CNNs can be used to compute the impact of SNPs on TF binding. « Mutation maps » can help to visualize such impact.

Mutations can lead to alternative splicing and be linked to genetic diseases



SpliceAI predict mutations affecting splicing

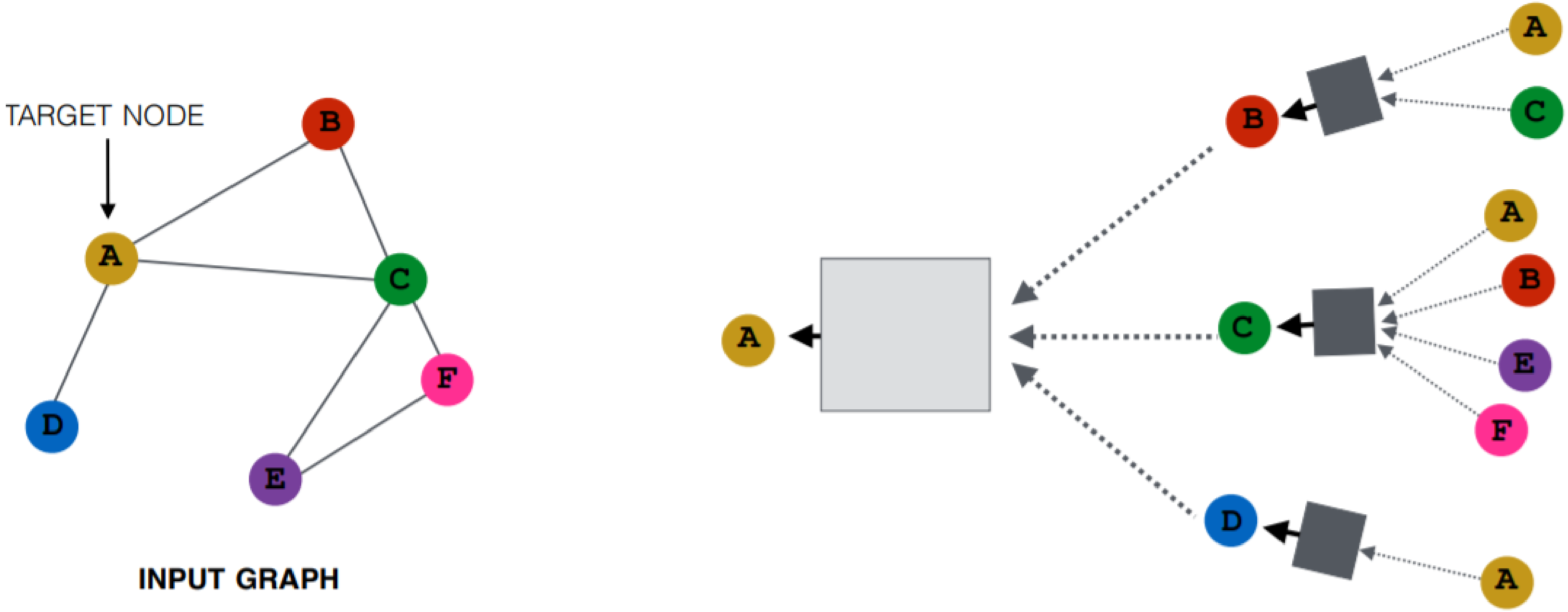
- Model trained using GENCODE-annotated pre-mRNA transcript sequences.



Illumina, Cell 2019.

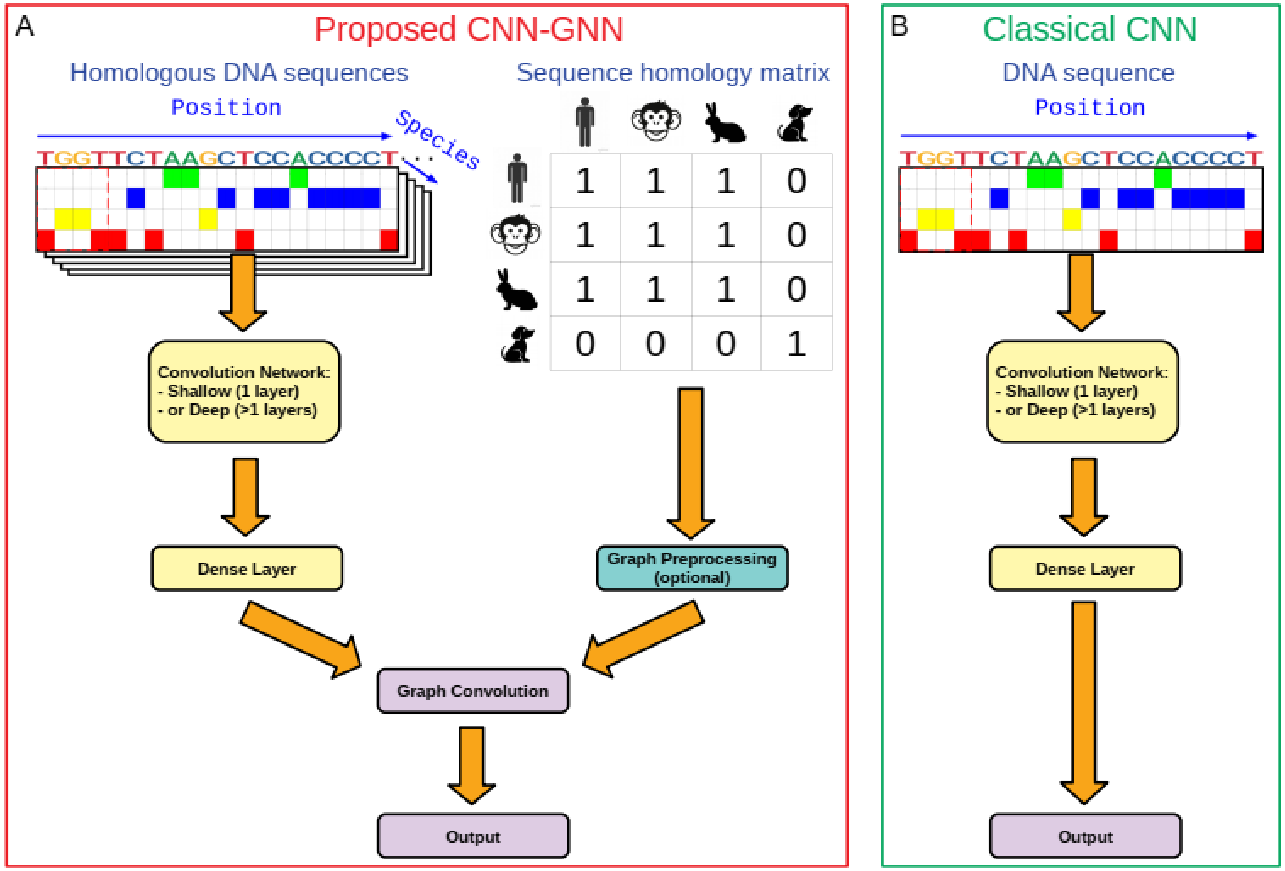
SEMI-SUPERVISED LEARNING

Graph Neural Network



Model homologuous species using graph neural networks

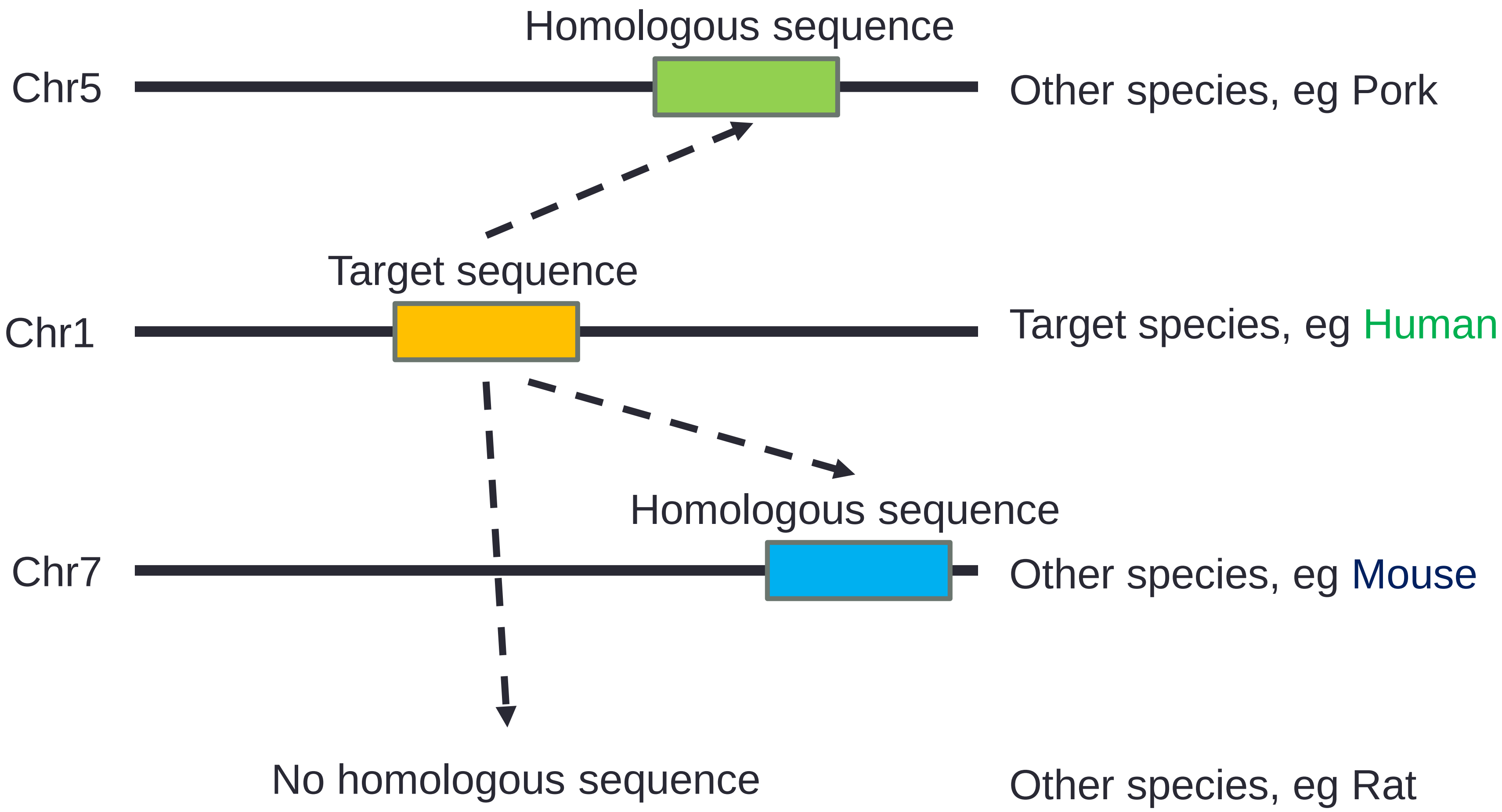
Global view of the model



	🧍	🐵	🐇	🐕
🧍	1	1	1	0
🐵	1	1	1	0
🐇	1	1	1	0
🐕	0	0	0	1

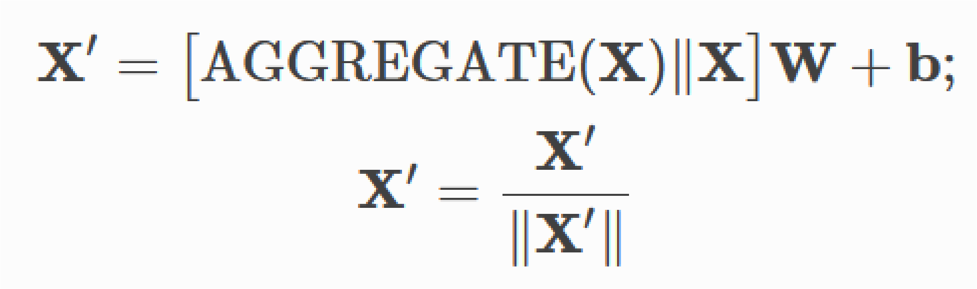
Figure 1. Sketch of the proposed semi-supervised model. A) A convolutional network within a graph neural network (so called CNN-GNN). B) Comparison with the classical convolutional network (CNN).

How to make the graph?



Graph Sage

- Graph Sage layer combines both aggregated features and original features:

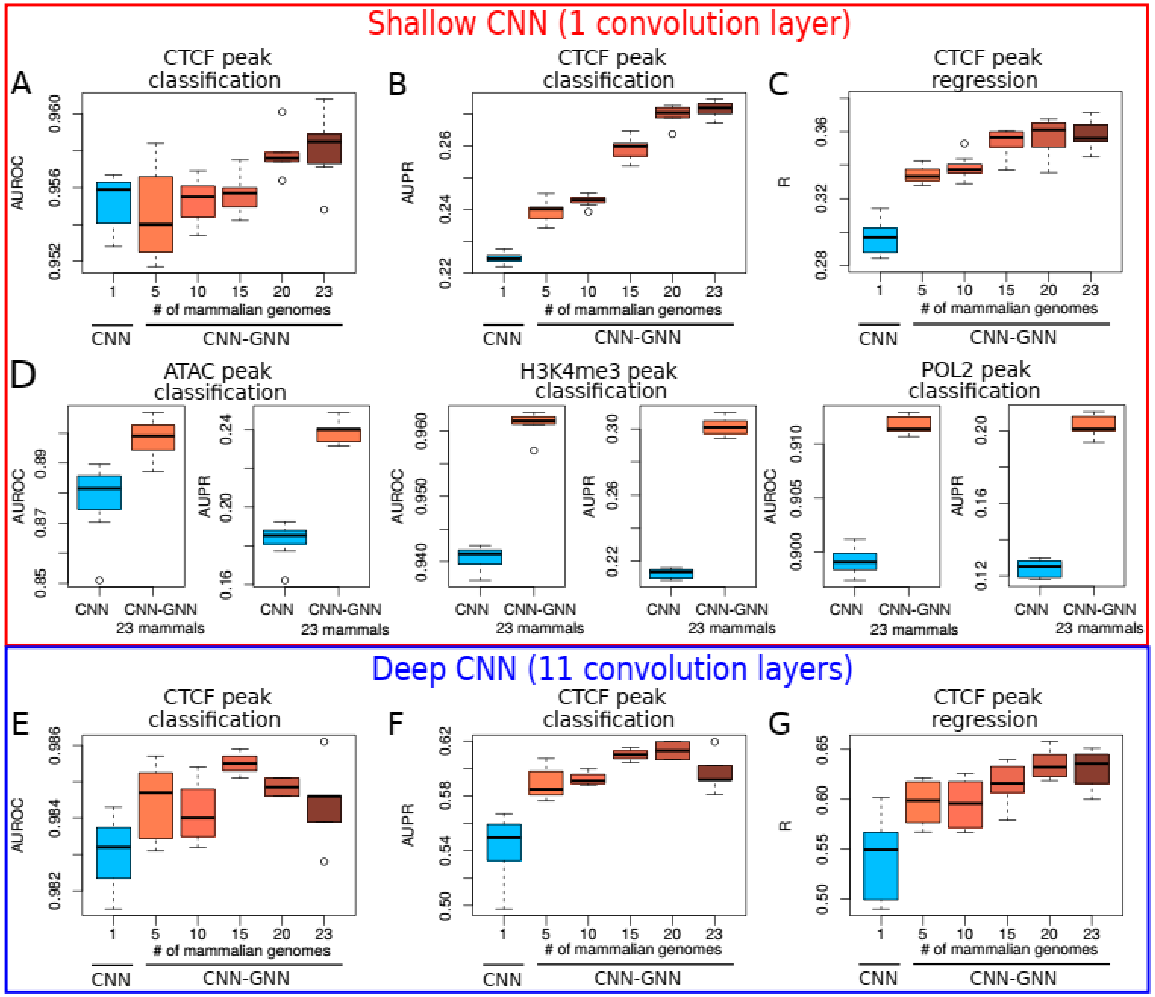


$$\mathbf{X}' = \left[\mathrm{AGGREGATE}(\mathbf{X}) \| \mathbf{X}\right] \mathbf{W} + \mathbf{b};$$

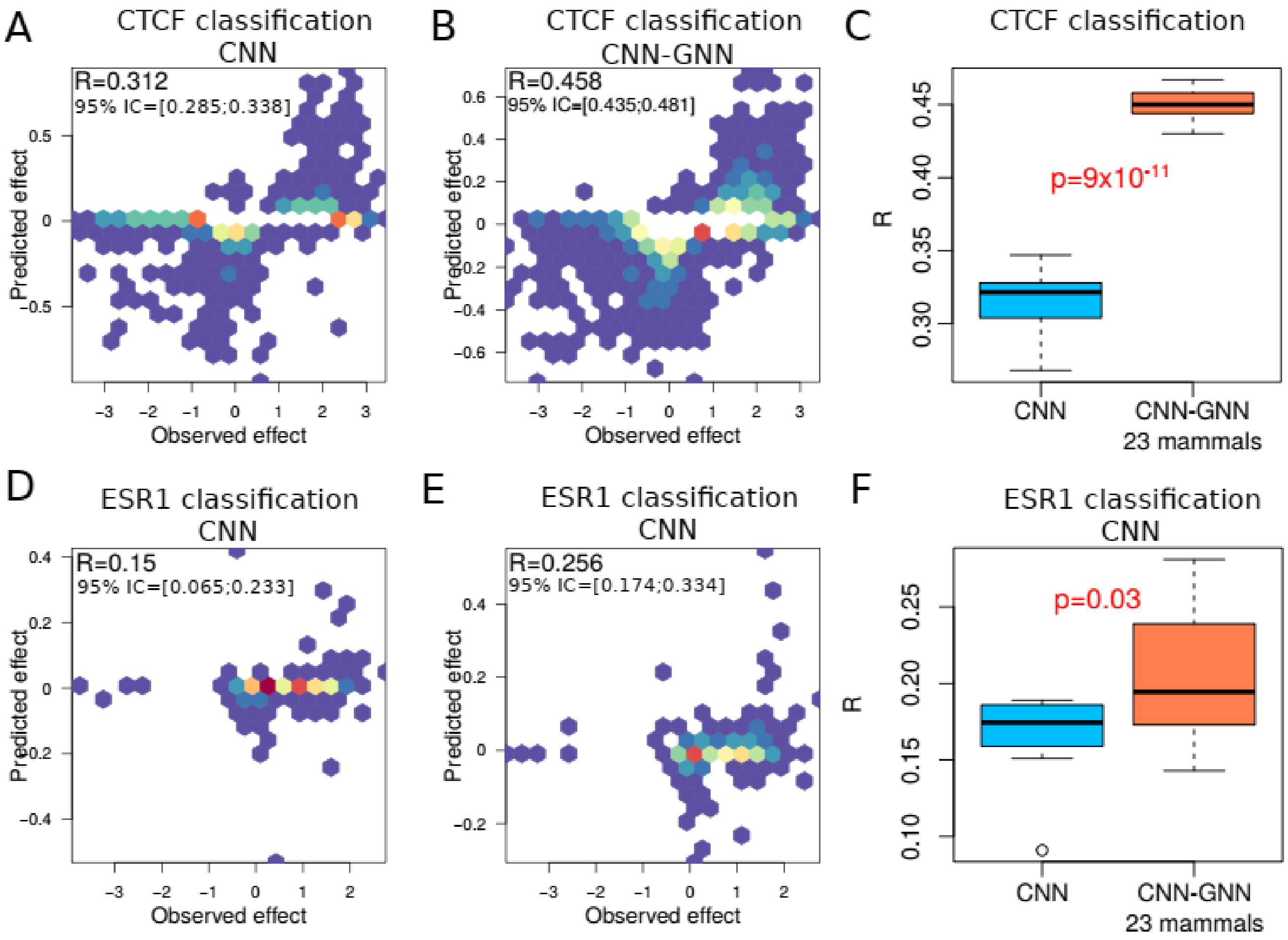
$$\mathbf{X}' = \frac{\mathbf{X}'}{\|\mathbf{X}'\|}$$

RESULTS

CNN-GNN improves baseline CNN



SNP effect prediction improved



THANKS FOR YOUR ATTENTION