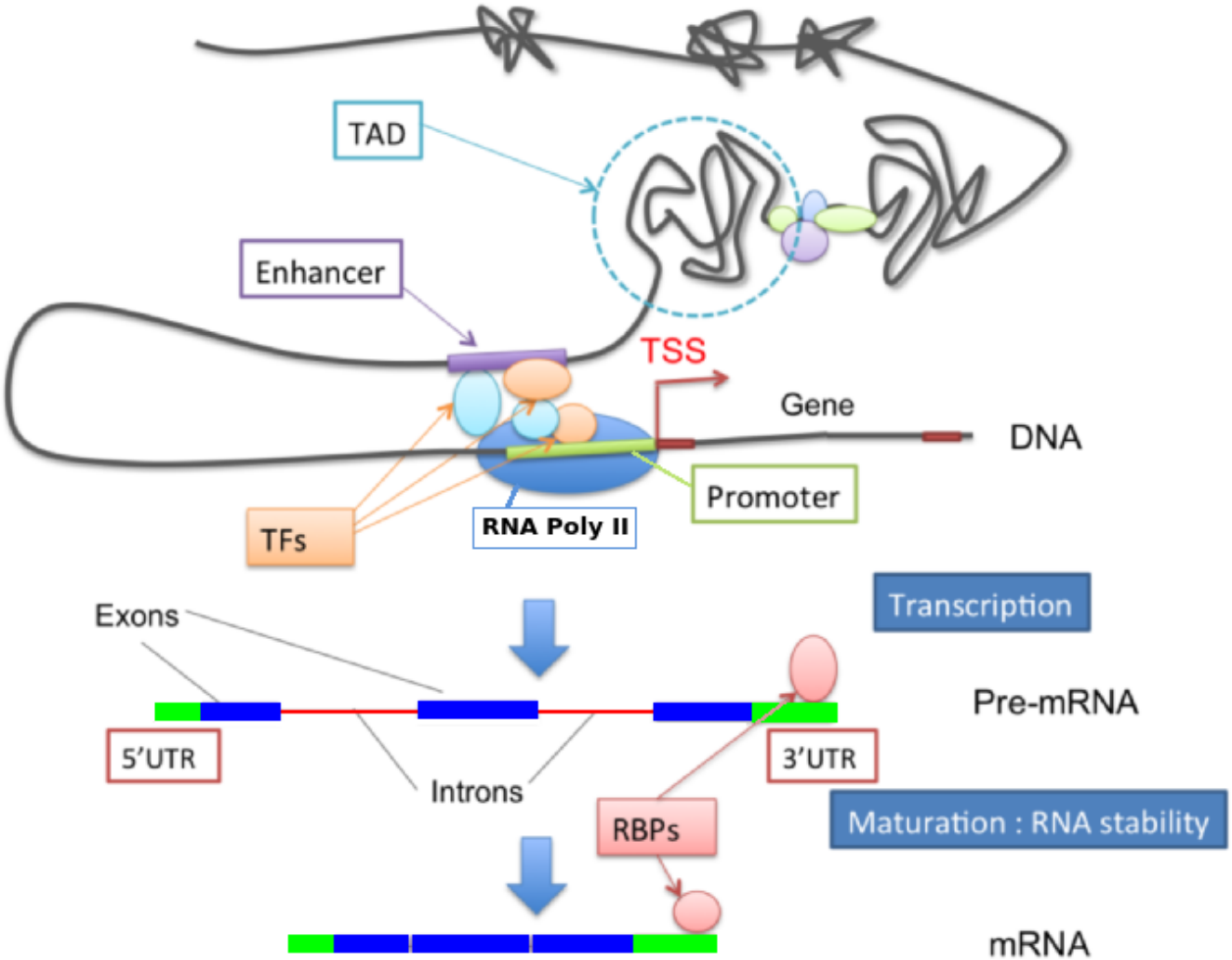
Combining genome features for gene expression modeling using convolutional network

May TAHA

IGMM - IMAG

December 14, 2018

Gene regulation



TFs = Transcription factors
RBPs = RNA Binding Proteins

Transcriptional regulations

Post-transcriptional regulations

Main elements to study regulation

DNA sequences:
- Regulatory regions
- Motifs

Experimental data:
- TF binding
- Epigenetics

Gene expression
- Count of mRNA

1- Predicting epigenetics based on DNA sequences

- DNA sequence can modulate the epigenome and ultimately gene expression [Quante & Bird Cell Biol (2016)]
- Specific DNA motifs can be associated to specific epigenetic marks [Whitaker & al. Nature (2015)]
- Predicting effects of non-coding variants with deep learning-based sequence model [Zhou & al. Nat.Methods (2015)]
- Convolution networks for quantifying the function of DNA sequences [Quang & al. NAR (2016)]

2- Predicting gene expression based on experimental data

- Regression analysis of combined gene expression regulation in acute myeloid leukemia [Li & al. PLoS CB (2014)]
- Combining transcription factor binding affinities with open-chromatin data for accurate gene expression prediction [Schmidt & al. NAR (2017)]
- Inference of transcriptional regulation in cancers [Jiang & al. Proc. Natl. Acad. Sci (2015)]

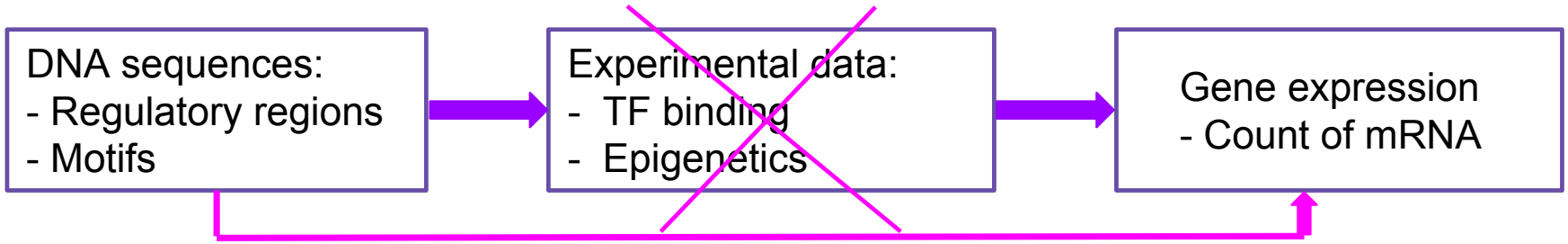
Limits of experimental data

These variables present biological and technical limits:

- Experimental data are cost and time consuming
- Not available for all conditions
- Do not capture regulation instructions that may lie at the sequence-level

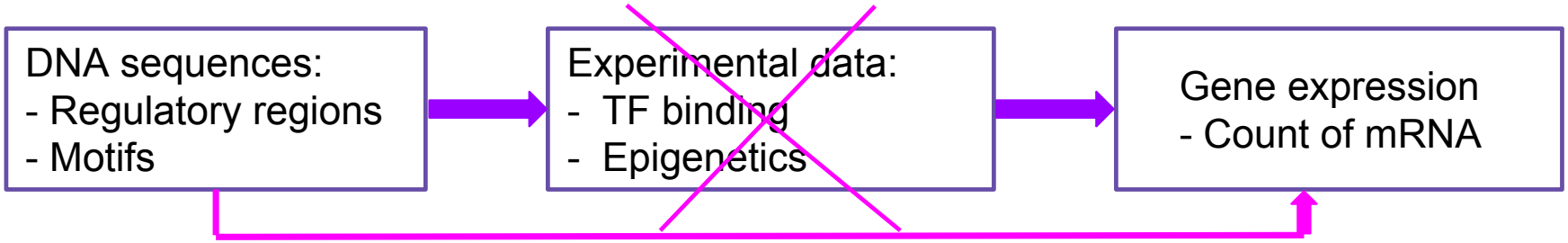
3- Predicting gene expression based on the DNA sequence

Our objective: Establish a model to predict and explain gene expression based only on DNA sequence level



3- Predicting gene expression based on the DNA sequence

Our objective: Establish a model to predict and explain gene expression based only on DNA sequence level



Concomitant works (2018)

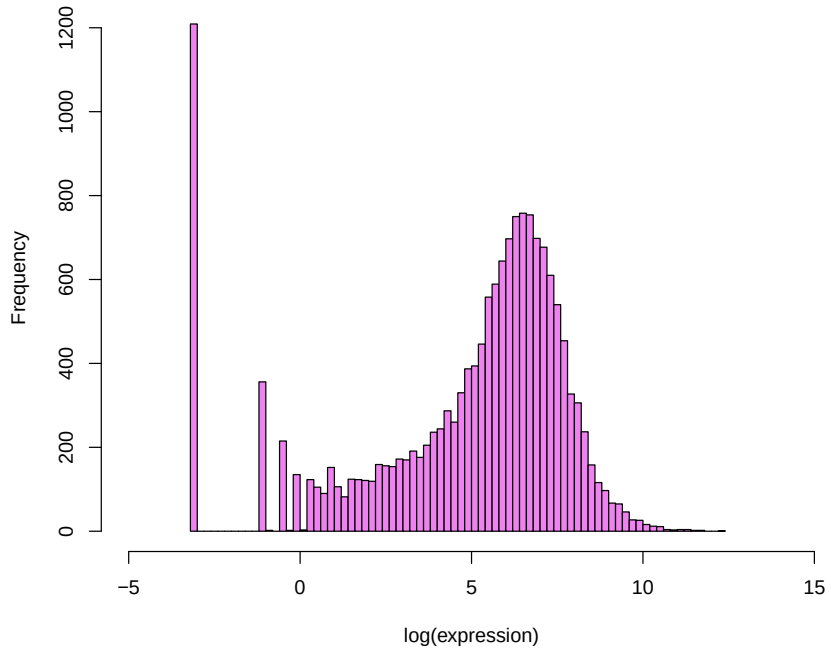
- Deep learning sequence-based ab initio prediction of variant effects on expression and disease risk [Zhou & al. Nature genetics (2018)]
- Sequential regulatory activity prediction across chromosomes with convolutional neural networks [kelley & al. Genome Research (2018)]
- Predicting mRNA abundance directly from genomic sequence using deep convolutional neural networks [Agarwal & Shendure BioRxiv (2018)]

Outline

1. Data: Gene expression and DNA sequence
   - Gene expression in cancer
   - Nucleotide compositions and Motifs
2. Summary of the penalized linear model
   - Article
   - Take home message
3. Convolution neural networks
   - Different networks
   - Convolution network architecture
   - Perspectives

Gene expression

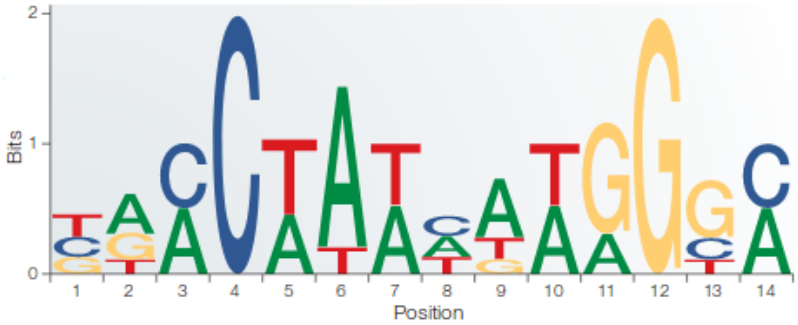
- RNA-seq data[1]
- 241 samples from 12 different cancers: AML, BRCA, ...



1- https://cancergenome.nih.gov/

Transcription factors binding sites: Motifs

JASPAR



$$W_{b,j} = log\left(\frac{P_{b,j}}{P(b)}\right)$$ base b, position j

Position Probability Matrix[2] PPM (P)

	1	2	3	4	5	6	7	8	9	10	11	12	13	14
A	0	0.5	0.5	0	0.375	0.875	0.5	0.375	0.625	0.5	0.25	0	0	0.5
C	0.375	0	0.5	1	0	0	0	0.375	0	0	0	0	0.25	0.5
G	0.25	0.375	0	0	0	0	0	0	0.125	0	0.75	1	0.625	0
T	0.375	0.125	0	0	0.625	0.125	0.5	0.25	0.25	0.5	0	0	0.125	0

Position Weight Matrix[3] PWM (W)

A	−1.93	0.79	0.79	−1.93	0.45	1.50	0.79	0.45	1.07	0.79	0.00	−1.93	−1.93	0.79
C	0.45	−1.93	0.79	1.68	−1.93	−1.93	−1.93	0.45	−1.93	−1.93	−1.93	−1.93	0.00	0.79
G	0.00	0.45	−1.93	−1.93	−1.93	−1.93	−1.93	−1.93	0.66	−1.93	1.30	1.68	1.07	−1.93
T	0.15	0.66	−1.93	−1.93	1.07	0.66	0.79	0.00	0.00	0.79	−1.93	−1.93	−0.66	−1.93

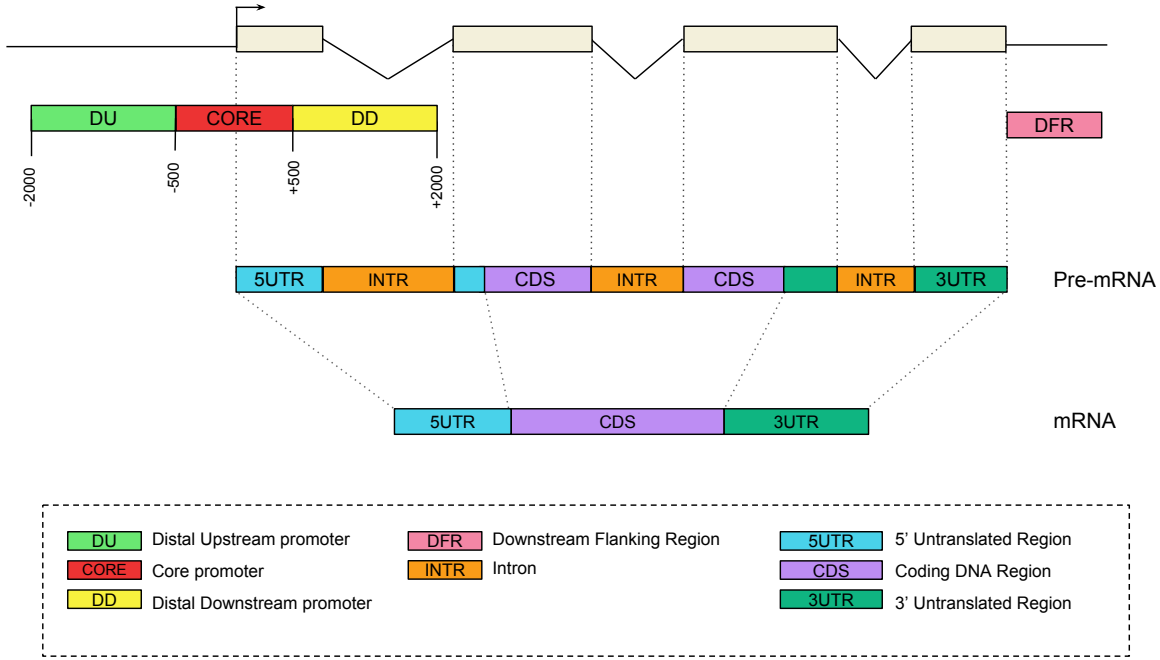
Computing score

$$Score(S, W) = \max_i \sum_{j=0}^{|W|-1} \log \frac{P(s_{i+j}|W_j)}{P(s_{i+j})}$$

2- [Mathelier & al. NAR (2016), Khan & al. NAR (2018)]

3- [Wyeth & al. Nat. Rev. Genet. (2004)]

Nucleotide compositions



$$percentage(N, s) = \frac{\sharp N}{|s|}$$

For each region:

- 4 nucleotides (A, C, G and T) and 16 di-nucleotides (CpG, CpA, ...)

Published work

PLOS COMPUTATIONAL BIOLOGY

RESEARCH ARTICLE

Probing instructions for expression regulation in gene nucleotide compositions

Chloé Bessière[1,2☯], May Taha[1,2,3☯], Florent Petitprez[1,2], Jimmy Vandel[1,4], Jean-Michel Marin[1,3], Laurent Bréhélin[1,4‡*], Sophie Lèbre[1,3,5‡*], Charles-Henri Lecellier[1,2‡*]

1 IBC, Univ. Montpellier, CNRS, Montpellier, France, **2** Institut de Génétique Moléculaire de Montpellier, University of Montpellier, CNRS, Montpellier, France, **3** IMAG, Univ. Montpellier, CNRS, Montpellier, France, **4** LIRMM, Univ. Montpellier, CNRS, Montpellier, France, **5** Univ. Paul-Valéry-Montpellier 3, Montpellier, France

☯ These authors contributed equally to this work.
‡ LB, SL, and CHL also contributed equally to this work.
* brehelin@lirmm.fr (LB); sophie.lebre@umontpellier.fr (SL); charles.lecellier@igmm.cnrs.fr (CHL)

Take home message

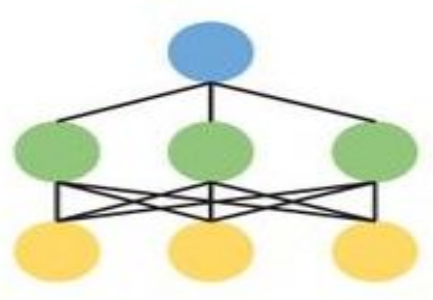
- A Lasso penalized linear model to predict gene expression based on nucleotide compositions in different regulatory regions
- DNA sequences contain information able to explain gene expression
- Sequence-level information is highly predictive of gene expression and in some occasions comparable to reference ChIP-seq data alone

Convolution neural network

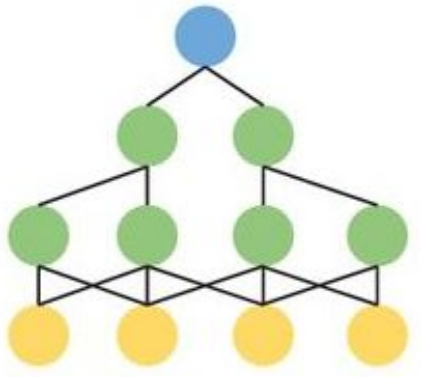
Types of network

There are different types of neural network. We used:

1. **Deep neural network**
   - More than two hidden layers
   - x_i: a binary or continuous vector
   - y_i: a binary or continuous scalar
   - Classification and regression



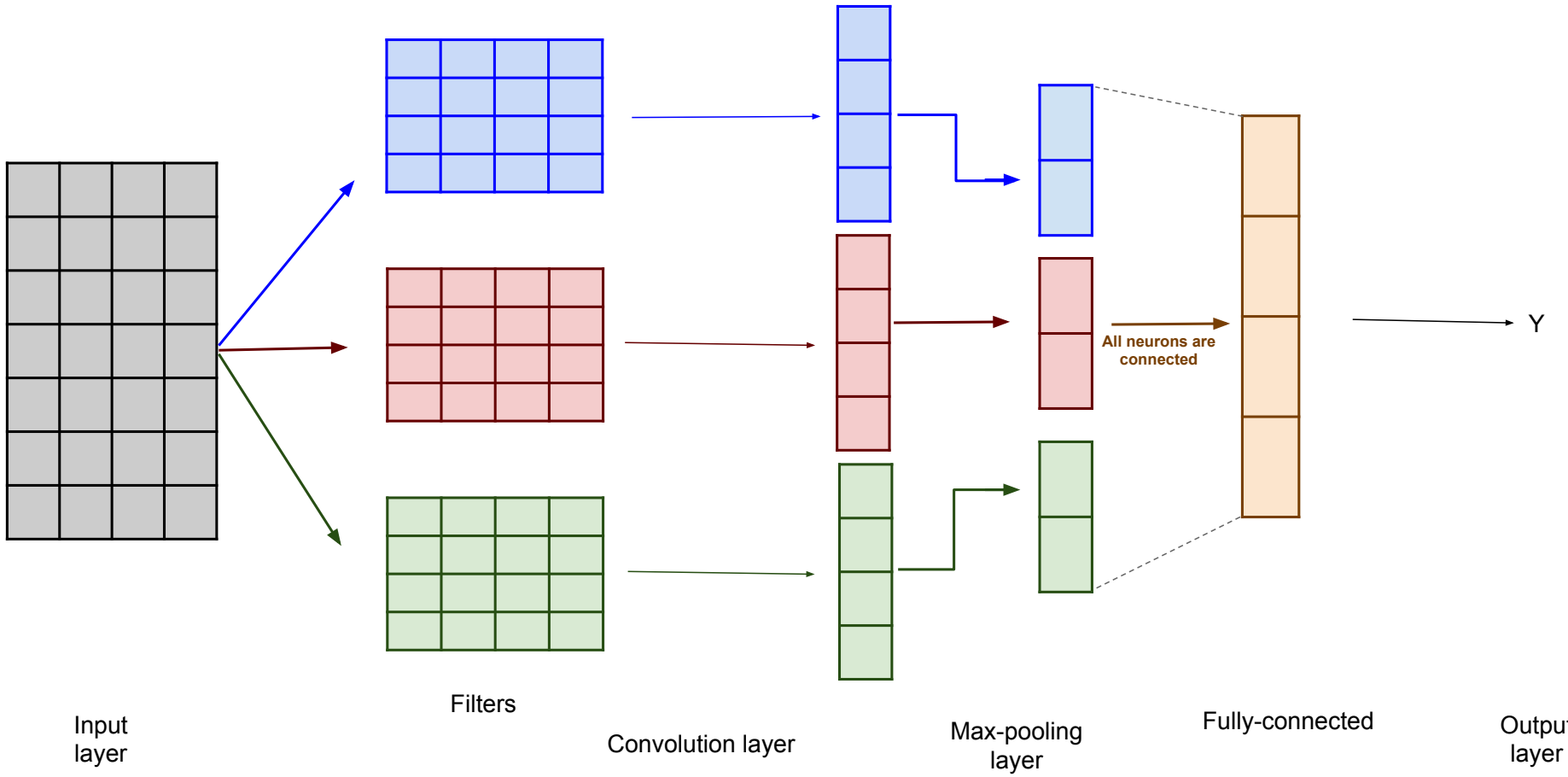
2. **Convolution neural network**
   - One or More layers
   - High number of neurons
   - X_i: a matrix (DNA Sequence, text, image)
   - y_i: a binary or continuous scalar
   - Classification and regression



Motivations

1. State of the art:
   - Convolution networks applied to DNA sequence is more and more used and developed over the years
   - Networks to predict epigenetics based on the sequences ([Quang & al. NAR (2016), Zhou & al. Nat.Methods (2015), . . .]
   - In 2018, predicting gene expression based on DNA sequence ([Zhou & al. Nature genetics (2018), Agarwal & Shendure BioRxiv (2018), . . .])
2. Using the DNA sequences as predictive variables instead of a summary of the sequence (scores and compositions)

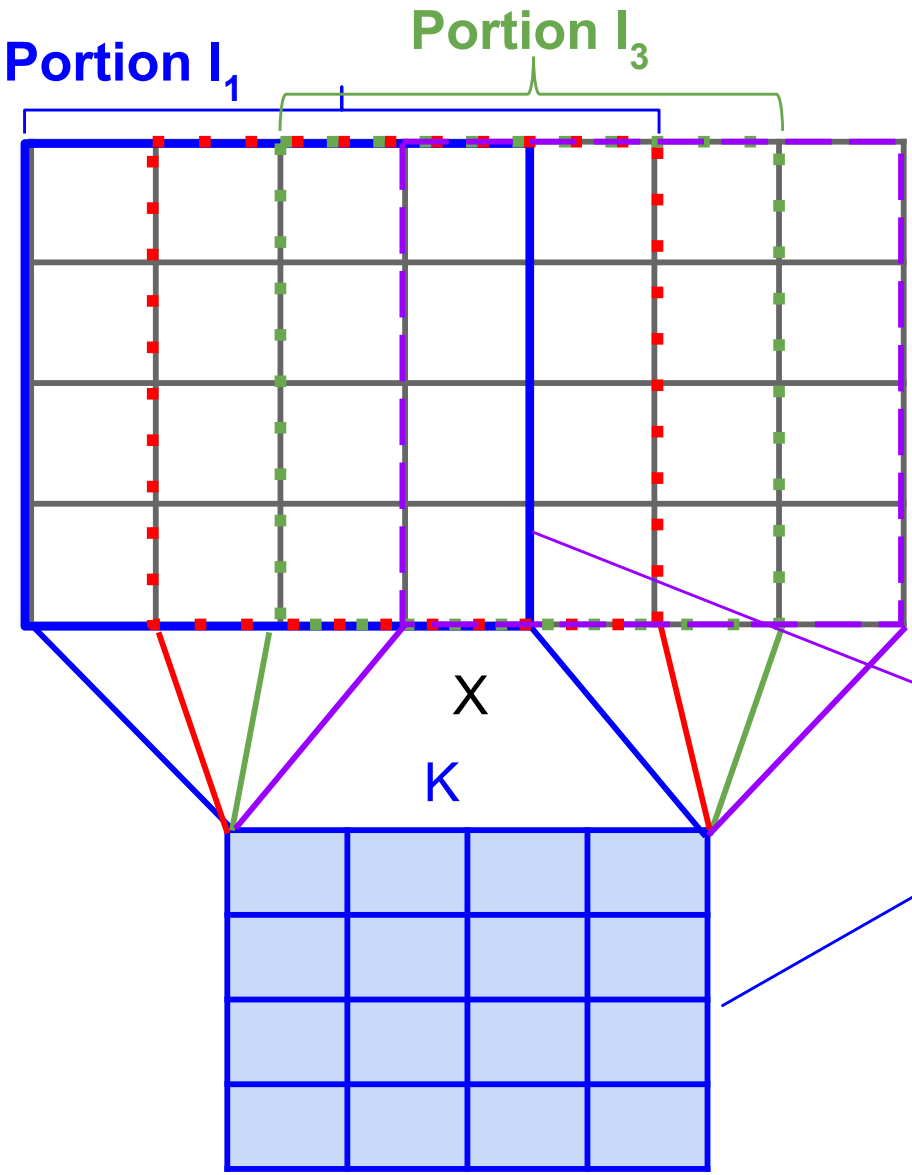
Convolution network



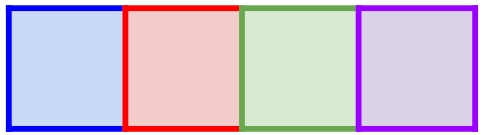
Gradient descent

Model weight estimations are obtained by the backpropagation algorithm of gradient descent optimization

Convolution layer



Feature map

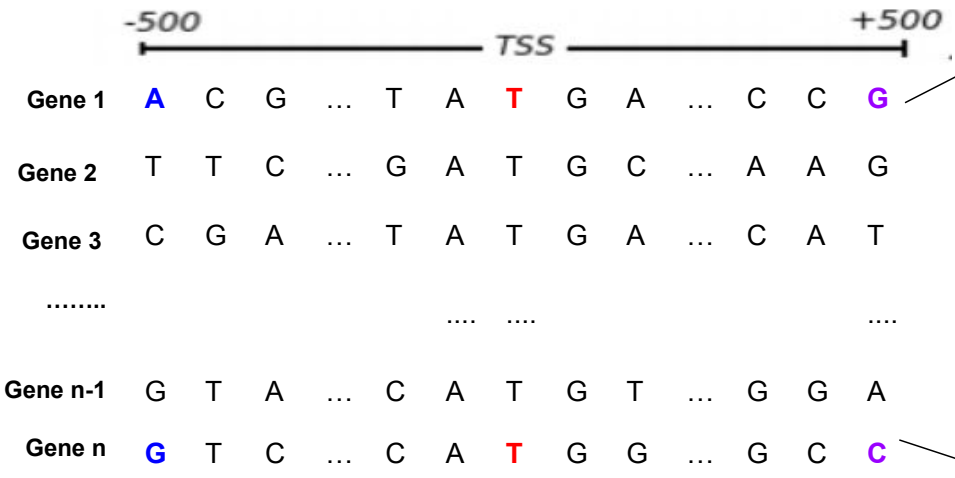


For I_1 (i.e. j=1):

$f(\sum_k \sum_n (I_{n,k+1} K_{n,w-k}))$ $k \in \{1,..., 4\}$ & $n \in \{1,..., 4\}$

Input layer

Promoter sequence -500/+500 b around TSS



Hot coding matrix for each gene

	-500	.....	0	.....	+500
A	1	.....	0	.....	0
C	0	.....	0	.....	0
G	0	.....	0	.....	1
T	0	.....	1	.....	0

	-500	.....	0	.....	+500
A	0	.....	0	.....	0
C	0	.....	0	.....	1
G	1	.....	0	.....	0
T	0	.....	1	.....	0

Hyperparameters

1. Number of convolution/pooling layers

Hyperparameters

1. Number of convolution/pooling layers
2. Type and window size of the pooling layer:
   - Maximum
   - Average
   - Window size can go from 1 to length of the output

Hyperparameters

1. Number of convolution/pooling layers
2. Type and window size of the pooling layer:
   - Maximum
   - Average
   - Window size can go from 1 to length of the output
3. Number of non-linear dense layers (ReLU: $f(x) = max(0, x)$ activation in general)

Hyperparameters

1. Number of convolution/pooling layers
2. Type and window size of the pooling layer:
   - Maximum
   - Average
   - Window size can go from 1 to length of the output
3. Number of non-linear dense layers (ReLU: $f(x) = max(0, x)$ activation in general)
4. Regularization:
   - Dropout with different probabilities
   - ℓ_1 and ℓ_2 regularization with different values of the λ

Hyperparameters

1. Number of convolution/pooling layers
2. Type and window size of the pooling layer:
   - Maximum
   - Average
   - Window size can go from 1 to length of the output
3. Number of non-linear dense layers (ReLU: $f(x) = max(0, x)$ activation in general)
4. Regularization:
   - Dropout with different probabilities
   - ℓ_1 and ℓ_2 regularization with different values of the λ
5. Training parameters: optimizer (Adam, RMSprop), number of epochs ....

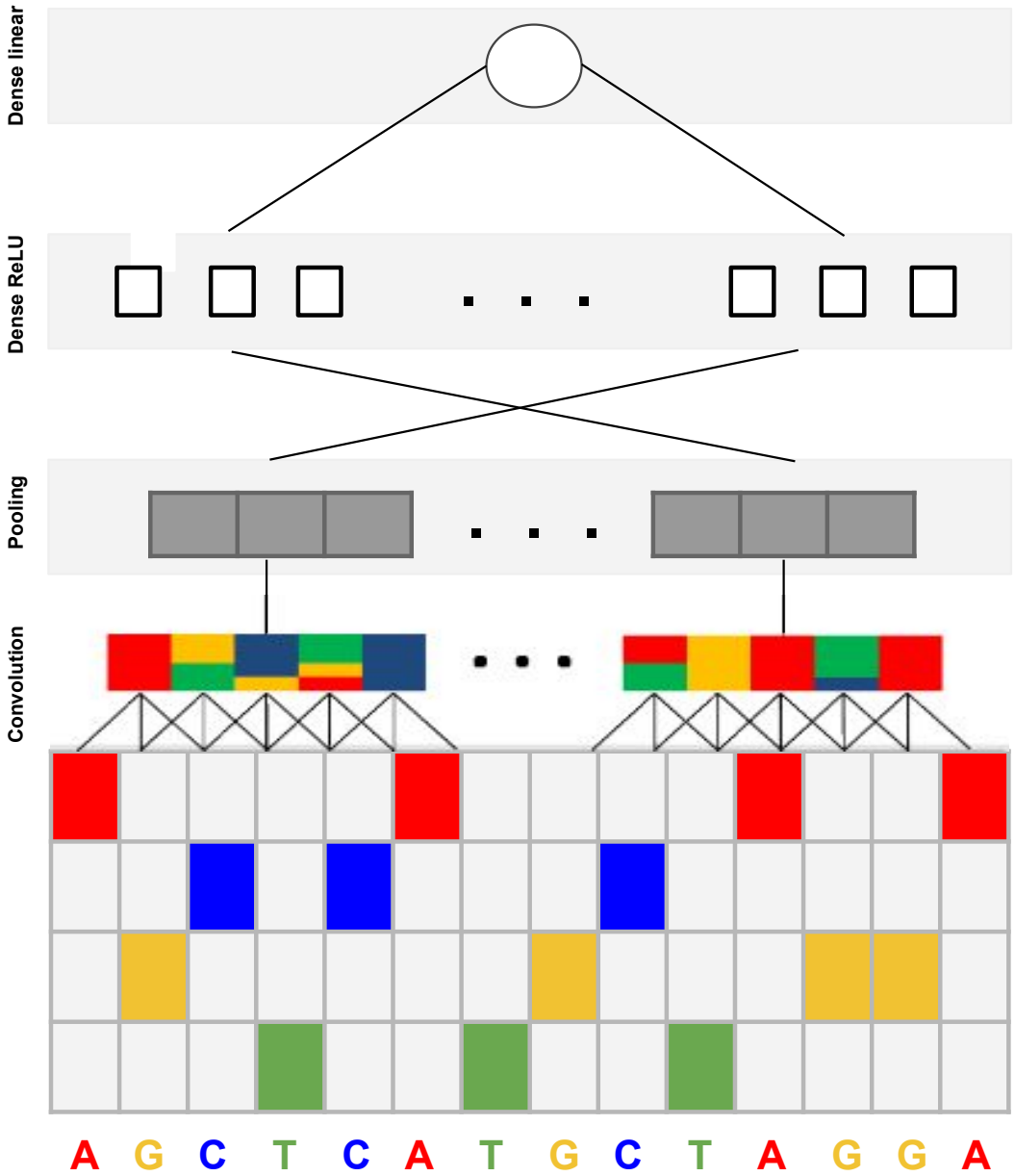
Parameter Optimization

Non-optimized hyperparameters

- Convolution: number of layers= 1, number of neurons = 550
- Training: number of epochs= 1000, optimizer = RMSprop

	Set of tested values
Initialisation Conv. weights	PPM, PWM, Random
Pooling layer	Maximum and Average With global, 10, 100 & 400 WS
Regularization	Drp = 0.4/ no drp
Neurons in ReLU Dense layer	2000, 200, 400, no layer

Architecture



Output layer (gene expression)
Dense layer: one neuron and linear activation

Dropout layer with p = 0.4

Dense layer: 200 neurons and ReLU activation function

Dropout layer with p = 0.4

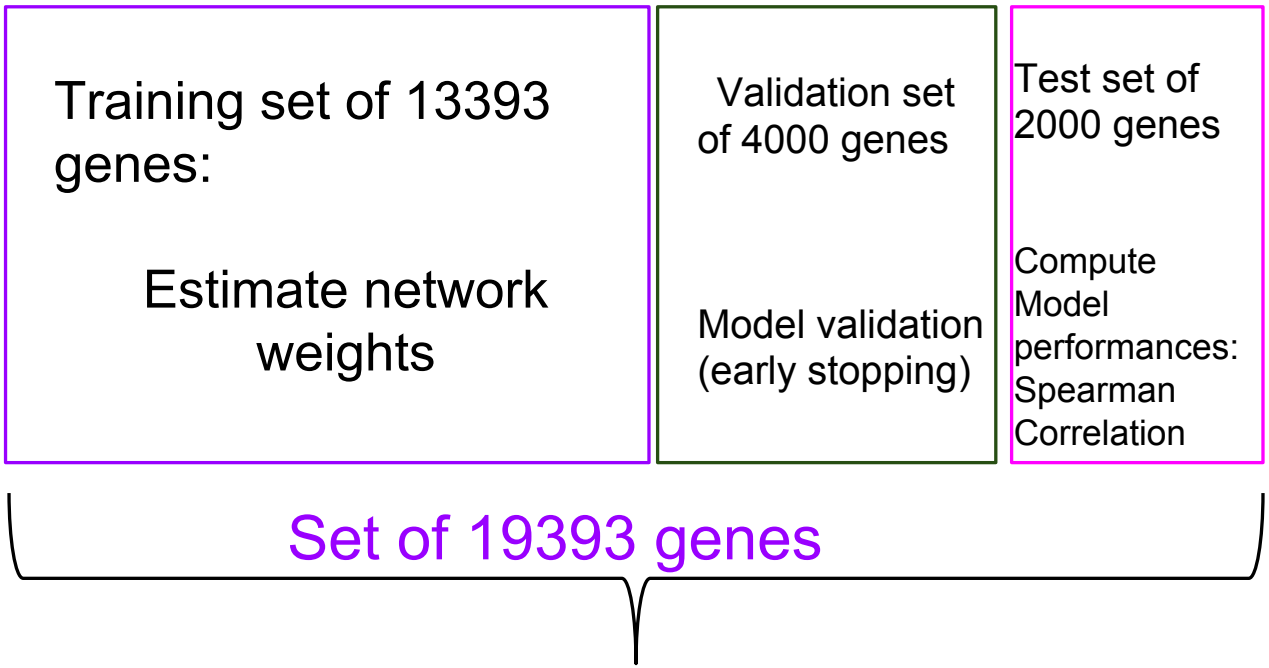
Maximum pooling layer: window size = 100

Convolution layer: 550 PPMs of length 15 b. ReLU activation function

Input layer: hot coding sequence of the CORE promoter (-500/+500 b around TSS.)

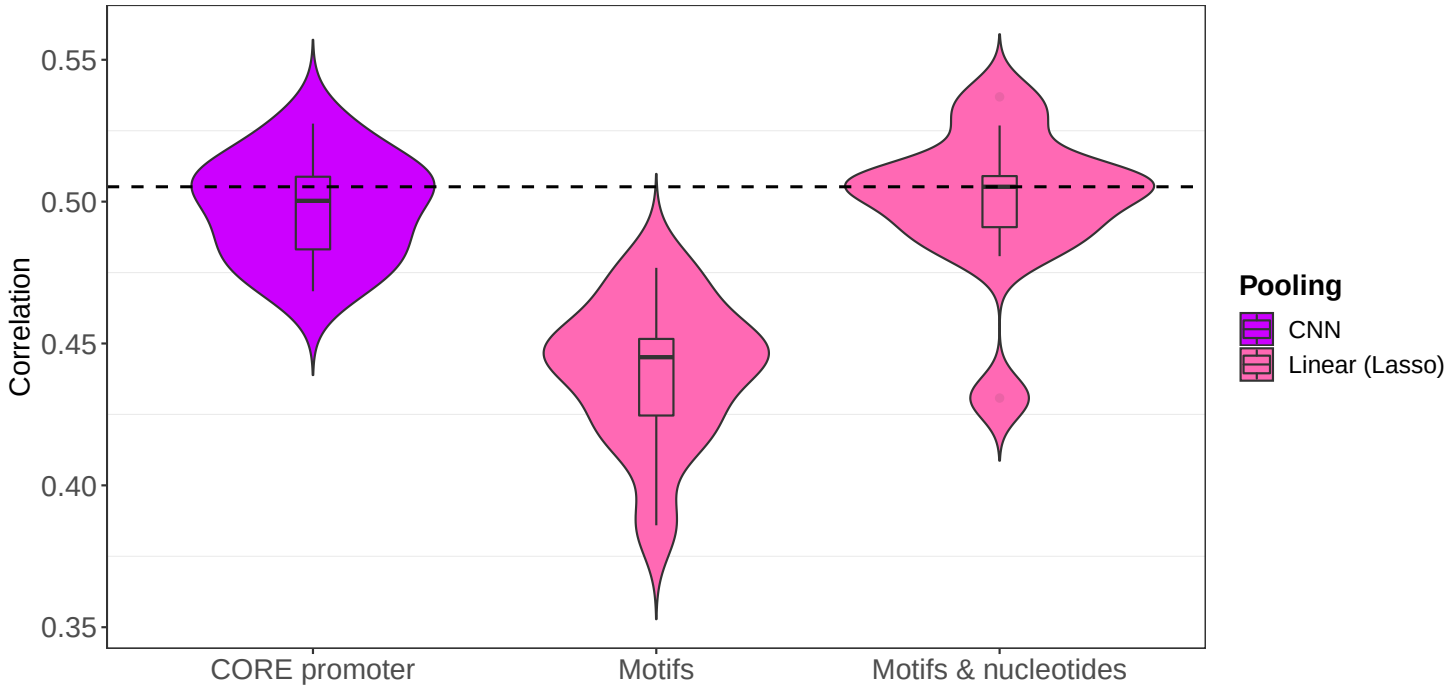
Validation procedure

1. On the set of genes



2. On the number of conditions:
   Only 12 conditions, one from each type of cancer, chosen randomly
   One model per patient

Results



- CNN model shows higher performances than Lasso penalized regression based only on motifs scores
- Similar results when fitting a linear model with both motifs and nucleotide composition in CORE promoter
- CNN models may capture the effect of both motifs and nucleotides

Limits and Perspectives

1. Hyperparameters were optimized by a manual search. Not considering dependencies.
   $\Rightarrow$ Optimized architecture using random search with the keras package "hyperopt" that select the model with lower prediction error

Limits and Perspectives

1. Hyperparameters were optimized by a manual search. Not considering dependencies.
   ⇒ Optimized architecture using random search with the keras package "hyperopt" that select the model with lower prediction error
2. Not enough input data to well estimate weights
   ⇒ Considering coding and non-coding genes

Limits and Perspectives

1. Hyperparameters were optimized by a manual search. Not considering dependencies.
   ⇒ Optimized architecture using random search with the keras package "hyperopt" that select the model with lower prediction error
2. Not enough input data to well estimate weights
   ⇒ Considering coding and non-coding genes
3. The sequence is limited to -500/+500 b
   ⇒ Consider larger sequence length This extension also may help to define interactions between different regions. Note: This may increase the number of parameters

Thank you for your attention