PROJET DE CLASSIFICATION DE SONDES MOLÉCULAIRES POUR LA DÉTECTION DE L'ARN IN SITU

Présenté par **Matéo Meynier**

Mai - Juillet 2020

Sous la tutelle de
Charles Lecellier

Encadré par
Thérèse Commes

01
INTRODUCTION

02
ETAT DE L'ART

03
CONCEPTION

04
RESULTATS

INTRODUCTION

EQUIPE
Biogenèse des ARNs

INSTITUT DE GÉNÉTIQUE MOLÉCULAIRE DE MONTPELLIER

EQUIPE
IGMM/LIRMM/IMAG
Regulations Génomiques Computationnelles

=> fournit les données biologiques (les sondes ADN dans mon cas)

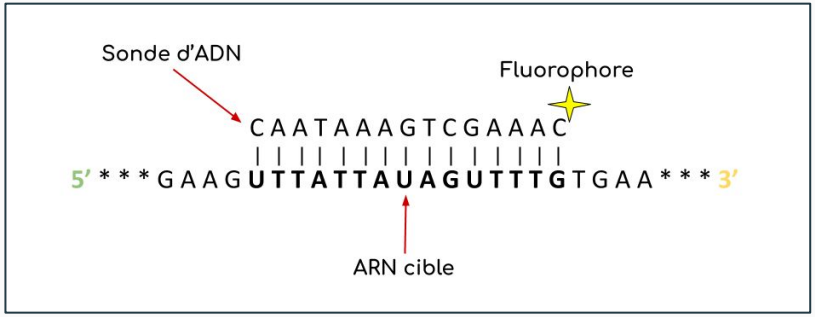
=> m'encadre pour concevoir le modèle de classification

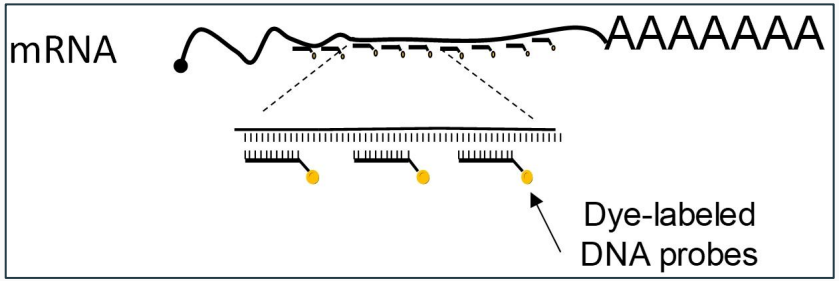
Single molecule fluorescence in-situ hybridization
smFISH

Technique de biologie moléculaire et d'imagerie basée sur l'hybridation et la complémentarité des bases nucléiques.

Utilisation de multiples sondes (séquences d'ADN) marquées avec un fluorophore
=> lien avec leurs séquences complémentaires

Objectif : quantifier et localiser des **molécules d'ARN cibles**

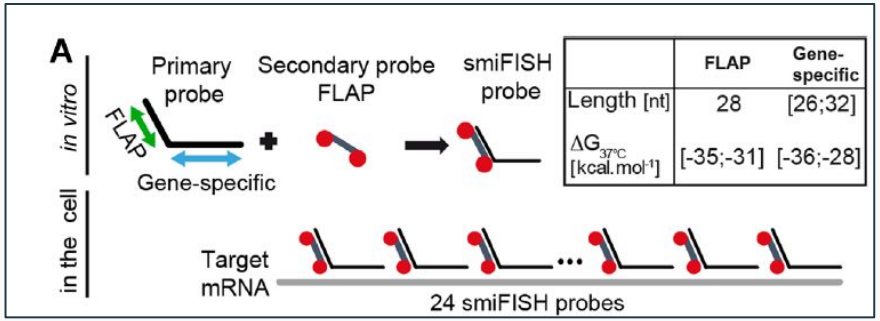


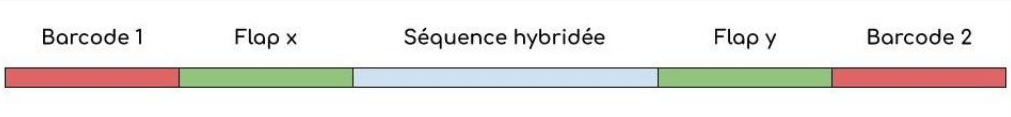


Alternative : smiFISH pour single molecule inexpensive FISH

- sondes primaires **non marquées**
- sondes secondaires **marquées**

Hybridation via la séquence FLAP





Source :
Tsanov N, Samacoits A, Chouaib R, et al. smiFISH and FISH-quant - a flexible single RNA detection approach with super-resolution capability. *Nucleic Acids Res.* 2016;44(22):e165.

PROBLEMATIQUE

Variation de la qualité du signal émis par les sondes ADN

Classifier les sondes

Bruit de fond	Pas de signal	Signal correct
0	1	2

Bruit de fond	Signal correct
0	1

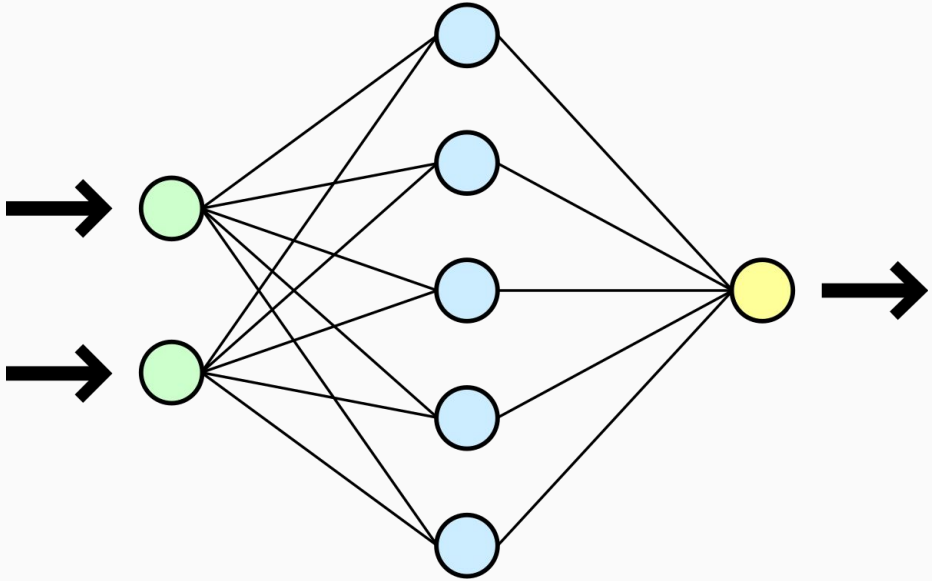
OBJECTIFS

- développer un **modèle d'apprentissage automatique** pour le problème de classification des sondes de smiFISH

Modèle : **Réseau de neurones à convolution (CNN)**

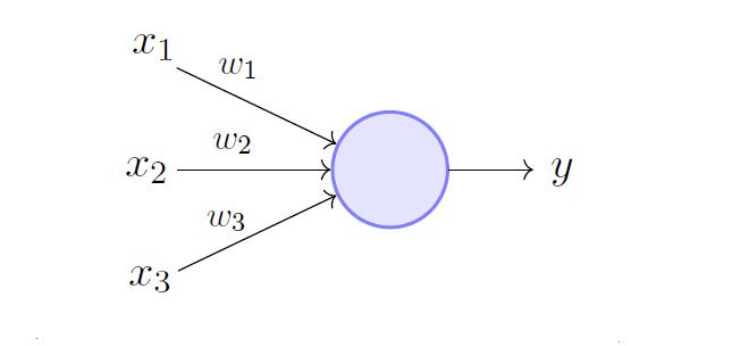
- extraire les caractéristiques que le modèle a découvert dans les séquences

- comparaison des caractéristiques extraites du CNN et celles de DExTER, modèle de régression logistique

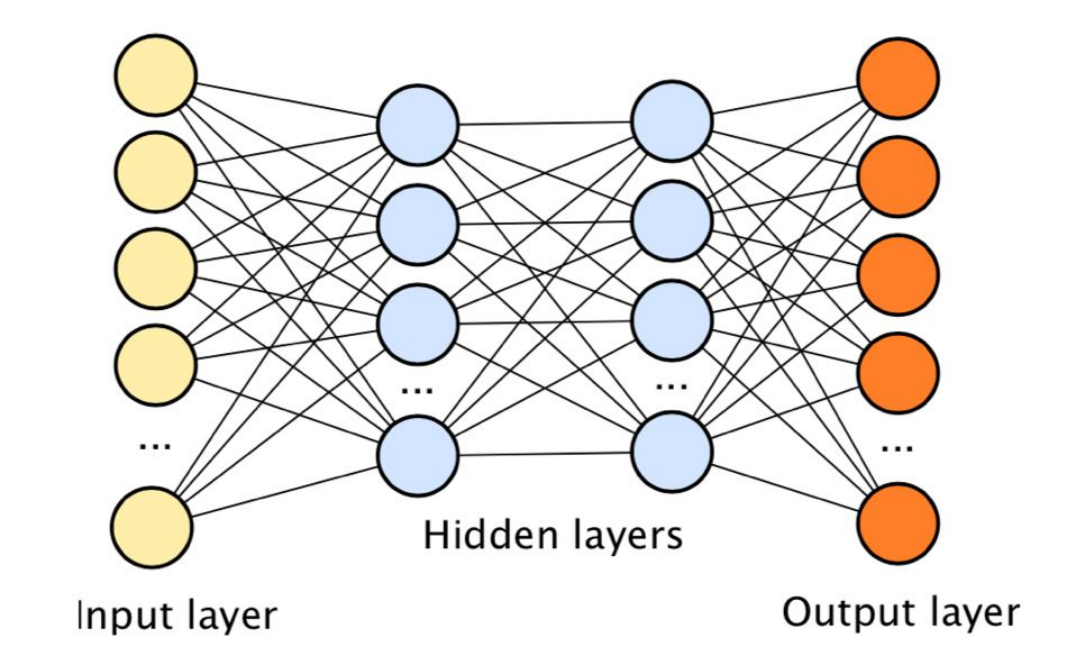


ETAT DE L'ART

NEURONE ARTIFICIEL OU PERCEPTRON

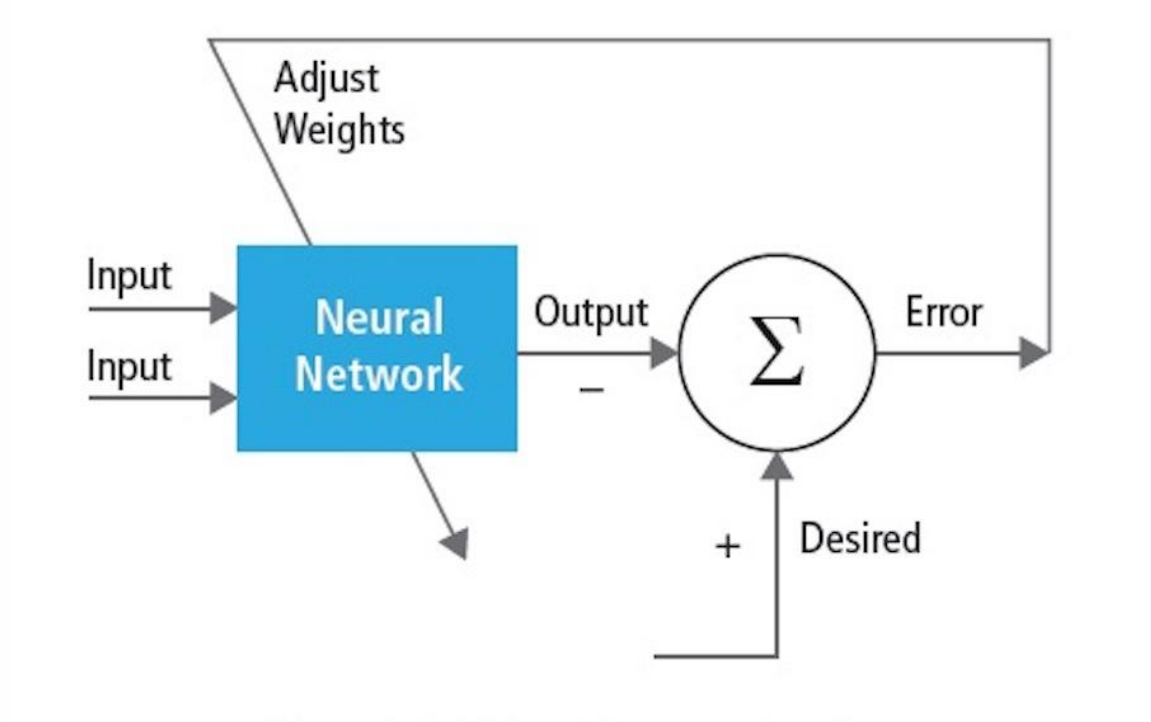


RESEAU DE NEURONES



Données d'entraînement

	gene	sig	
0	NFKBIA_0	1	TCCCATGGGCAGTATCGCTTTGA
1	NFKBIA_1	1	TCCCATGGGCAGTATCGCTTTG
2	NFKBIA_2	1	TCCCATGGGCAGTATCGCTTTG
3	NFKBIA_3	1	TCCCATGGGCAGTATCGCTTTGA
4	NFKBIA_4	1	TCCCATGGGCAGTATCGCTTTGA
...	...	...	
96300	ZNF213-AS1_56	0	GTGCAGGGCACTTCCGCTGTAC
96301	ZNF213-AS1_57	0	GTGCAGGGCACTTCCGCTGTAC
96302	ZNF213-AS1_58	0	GTGCAGGGCACTTCCGCTGTAC
96303	ZNF213-AS1_59	0	GTGCAGGGCACTTCCGCTGTAC
96304	ZNF213-AS1_60	0	GTGCAGGGCACTTCCGCTGTAC

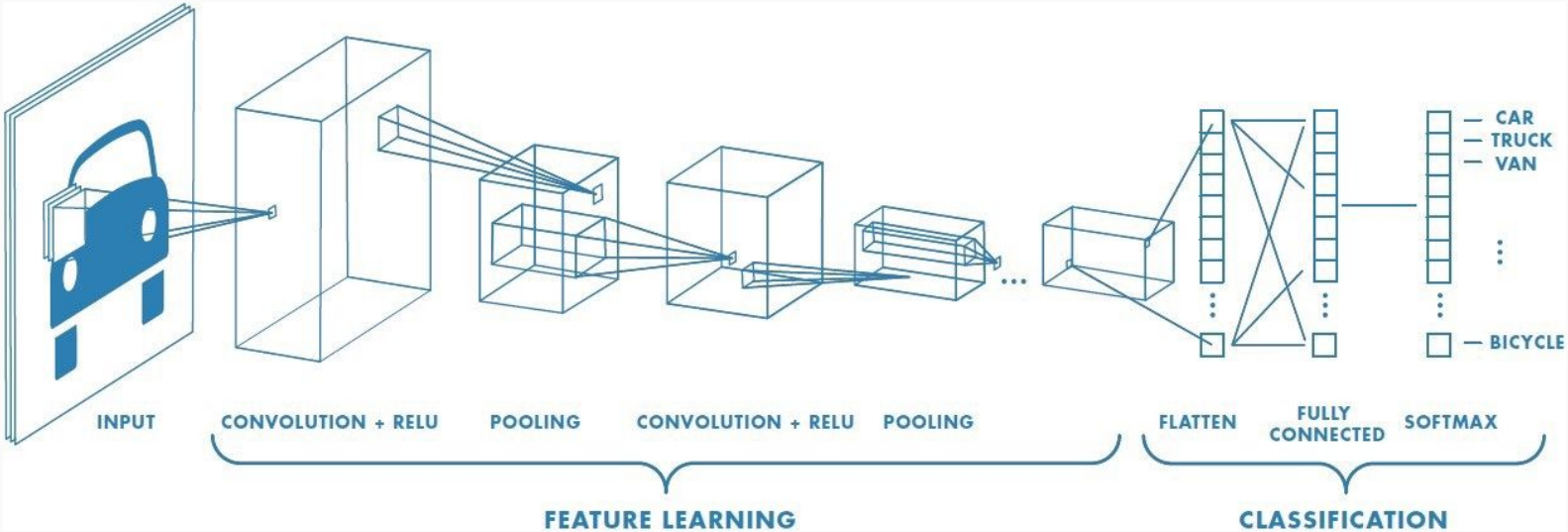


Opération de convolution

application d'un filtre pour détecter des caractéristiques

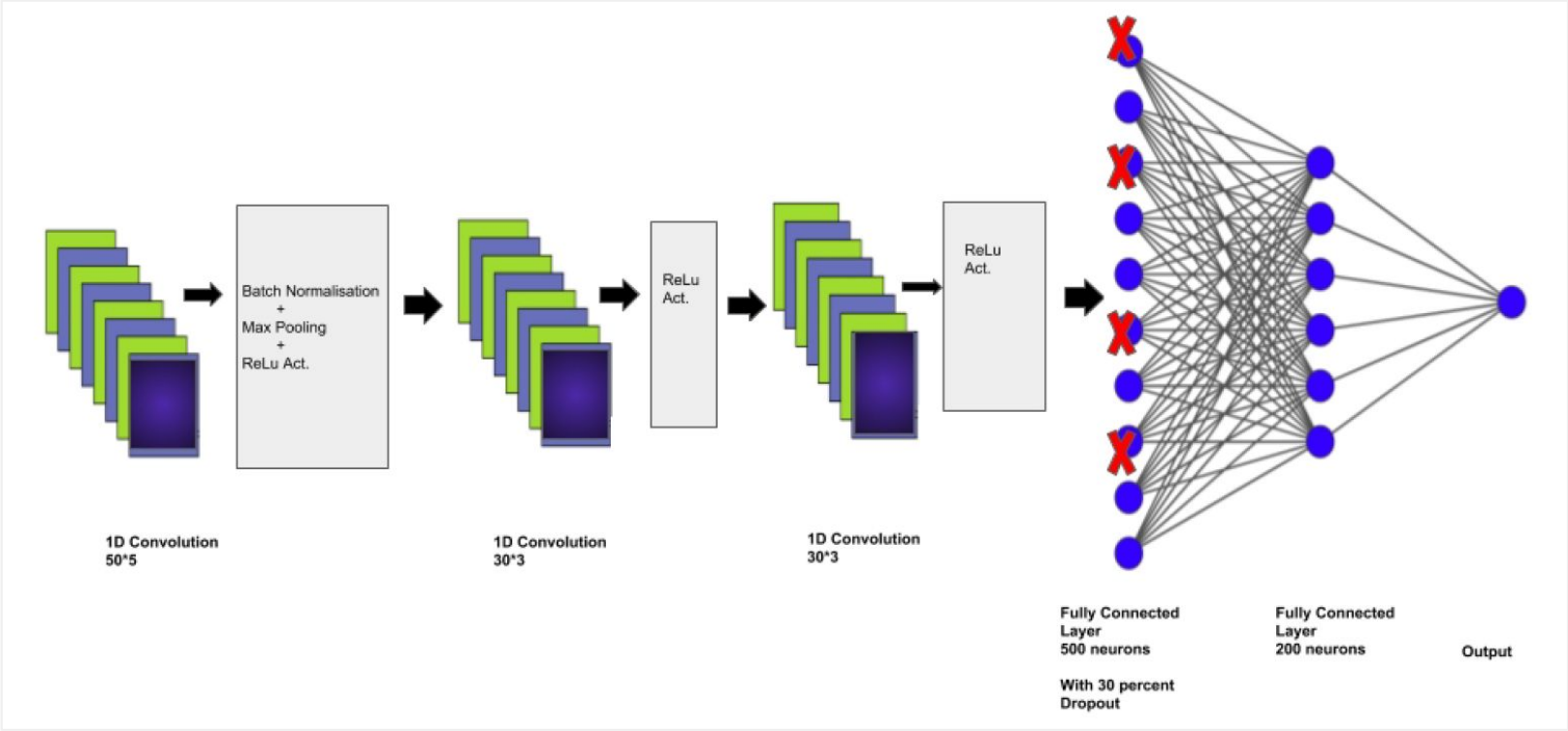
Opération de pooling

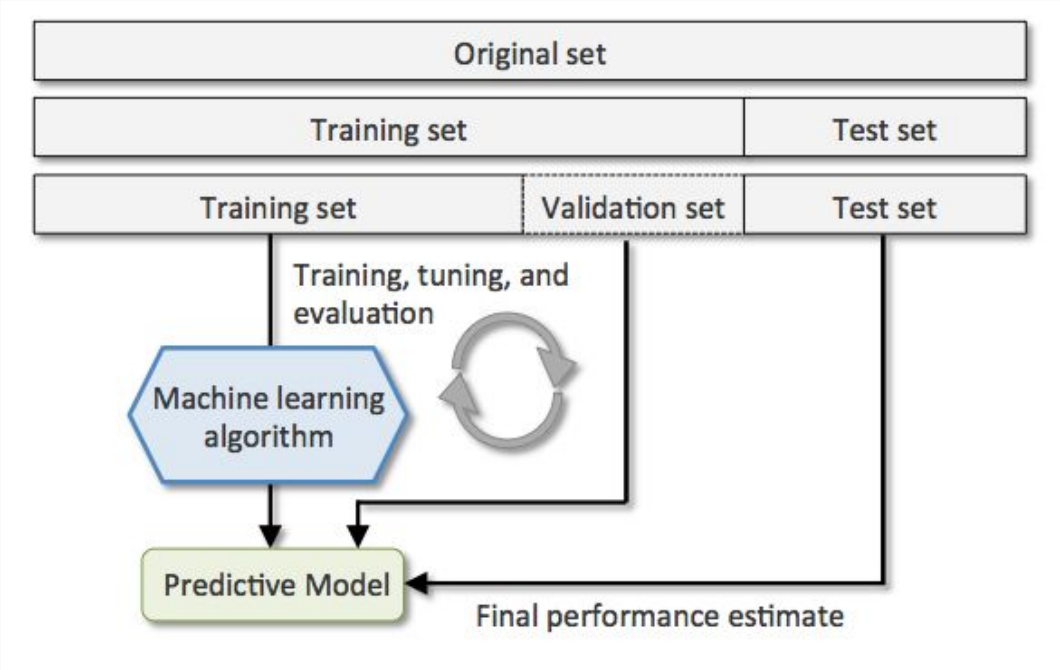
Réduire l'image tout en conservant les caractéristiques détectées

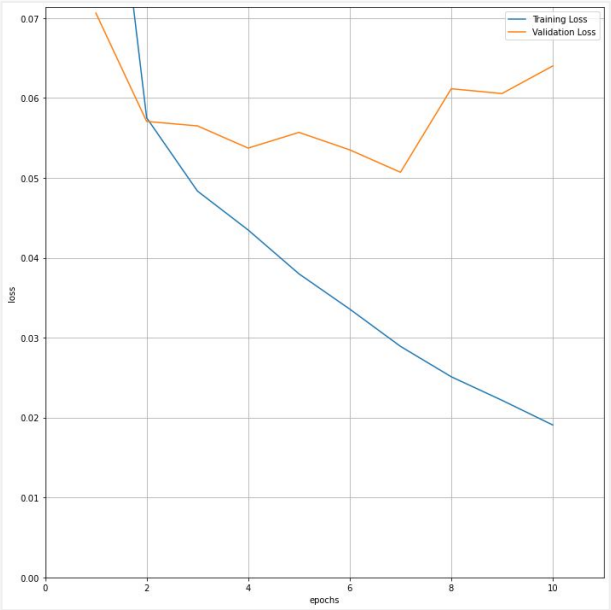


CONCEPTION

CONCEPTION I. ARCHITECTURE DU CNN




Original set
Training set
Test set
Training set
Validation set
Test set
Training, tuning, and evaluation
Machine learning algorithm
Predictive Model
Final performance estimate


Training Loss
Validation Loss
loss
epochs
0.07
0.06
0.05
0.04
0.03
0.02
0.01
0.00
0
2
4
6
8
10

RÉSULTATS

RÉSULTATS I. PREMIERS RÉSULTATS

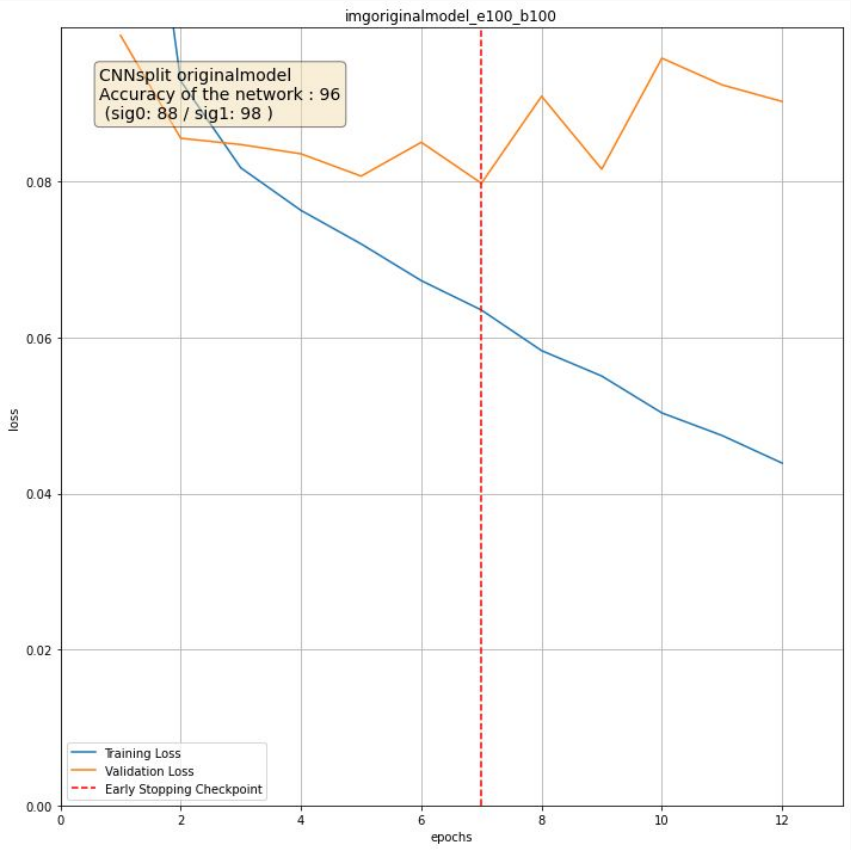
Données fournies au CNN

96 305 exemples

{nom du gène, signal, séquence}

Set d'entrainement :
76 062 seqs (~80%)

Set de test :
20 243 seqs (~20%)



RÉSULTATS I. PREMIERS RÉSULTATS

F1-score : moyenne de la spécificité d'un modèle et de sa sensibilité

Sensibilité : probabilité de bien détecter un exemple positif

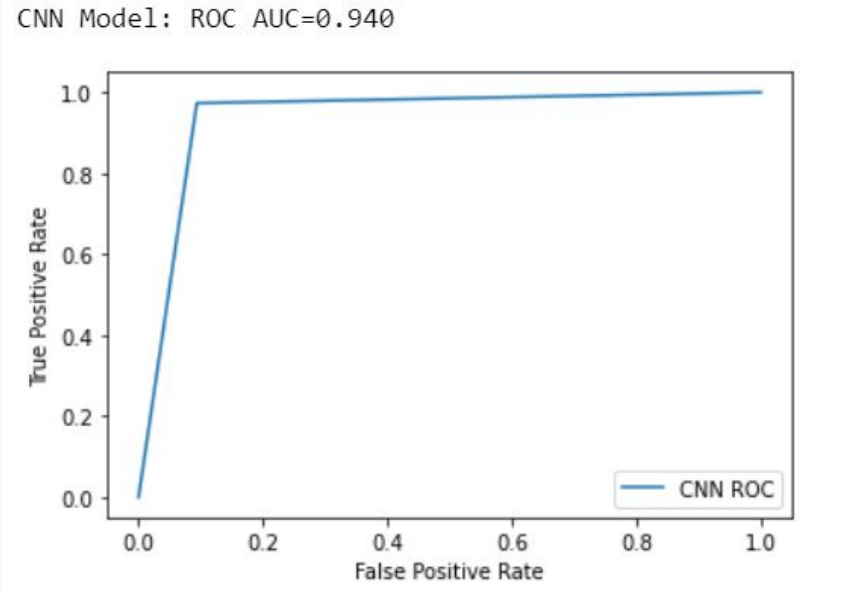
Spécificité : probabilité de ne pas détecter un exemple négatif comme positif

Courbe ROC : mesure de la performance d'un classificateur binaire

True positive rate : fraction des positifs qui sont effectivement détectés
False positive rate : fraction des négatifs qui sont incorrectement détectés

```
Accuracy (classification_report) :  0.9631898655313846

 Signal0 f1-score (classification_report) :  0.8810203054203726
Signal1 f1-score (classification_report) :  0.9782268218530233
```



RÉSULTATS II. RÉSULTATS D'ÉLODIE SIMPHOR

Méthode DEXTER

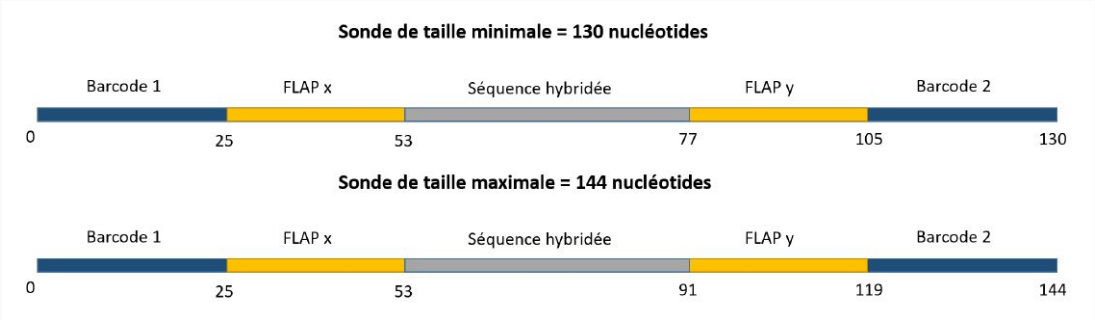
Domain

Exploration To Explain gene Regulation

- Identifie des paires [k-mers,regions] où il existe une corrélation entre la classe du signal qualité et la fréquence du k-mer dans la région définie de chaque sonde

- Ensemble des paires identifiées => permet la prédiction du signal

- Basée sur une méthode de régression logistique



Variables (séquences)	AA	AG	CTAC	GCT	GGC	GTAT	TA	TAT	TGAG	TGAG
Position dans la séquence	1-10	0	10	0-20	10	12-100	0-12	91-100	71-110	91-110
Valeur du coefficient	-1,28	-0.56	-4.45	-0.90	-1.56	-34.12	-0.64	-1.64	-8.86	-11.23

Variables (séquences)	GA	GCTG	GGC	GTAT	TCC	TCC	TCCCT	TCT	TCTAG	TGT
Position dans la séquence	73-131	12	12-131	12-108	12	109-131	12-120	0-12	97-131	1-12
Valeur du coefficient	-14.70	-2.73	-5.89	-58.43	-3.39	0.35	-25.73	1.48	-40.28	-4.25

Conclusion :
Variables discriminantes au niveau des **barcodes** principalement et des **FLAPs**

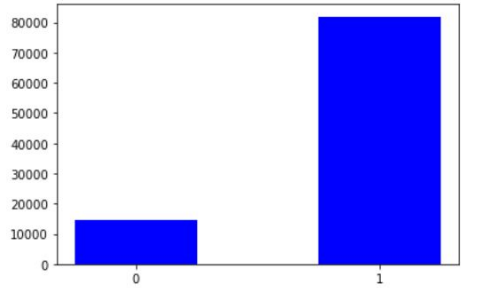
RÉSULTATS III. NOUVEAUX RÉSULTATS

Pour contrer le sur-apprentissage , découpage des données de sorte que le CNN soit testé sur des barcodes sur lequel il ne s'est pas entraîné

Or une paire de barcode (barcodes 1 et 2) est **spécifique** à un gène

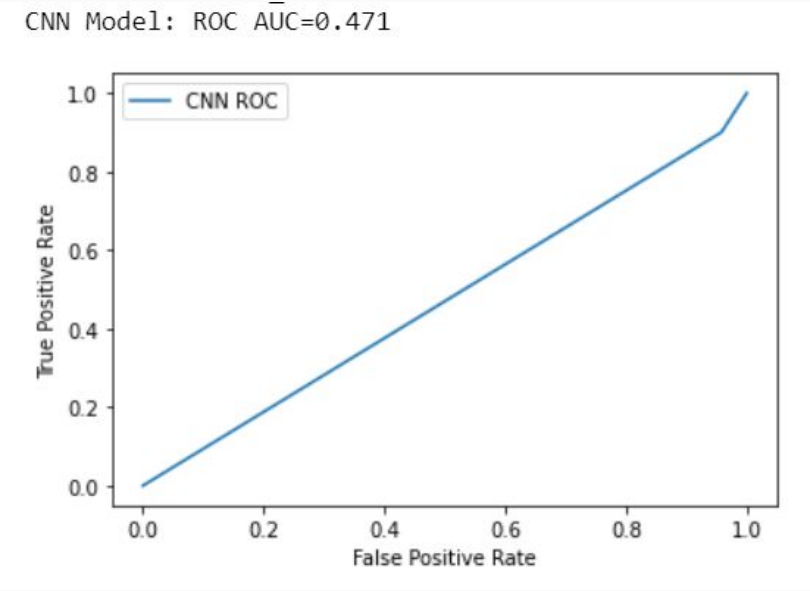
=> **découpage** en fonction des **gènes**
80% des gènes pour le set d'entraînement
20% des gènes pour le set de test

```
--- SIG1 :  85.06619593998235 % ---
--- SIG0 :  14.933804060017653 % ---
--- SIG1 number :  81923 ---
--- SIG0 number :  14382  ---
```




```
Accuracy (classification_report) :  0.7512539899680802

 Signal0 f1-score (classification_report) :  0.05541125541125541
Signal1 f1-score (classification_report) :  0.856767756334515
```



RÉSULTATS IV. RÉÉQUILIBRAGE DES CLASSES

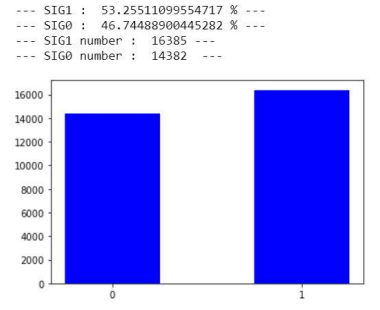
Pour rééquilibrer les classes des signaux :

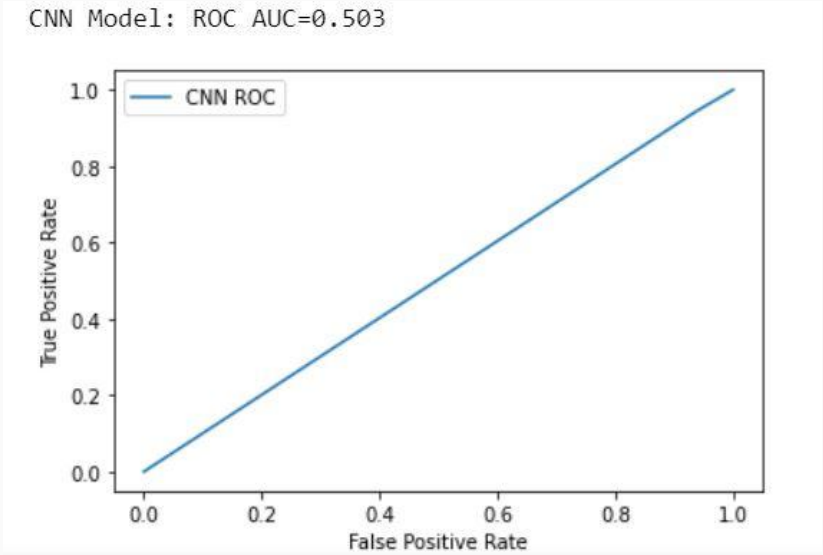
- enlever des données de classe 1 manuellement
- appliquer des poids aux classes via la fonction de coût
- utiliser l'échantillonneur WeightedRandomSampler pour augmenter virtuellement les données de classe 0

```
--- Résultats du set de test ---

Accuracy (classification_report) :  0.5015178143473399

 Signal0 f1-score (classification_report) :  0.3807066296149266
Signal1 f1-score (classification_report) :  0.5828877005347594
```





CONCLUSION

⇒ raison de l'échec du CNN : le **manque** de données et plus précisément de **barcodes**

Pour 90 000 séquences ⇒ **87 paires de barcodes différentes = pas assez d'exemples**

Or un réseau de neurones nécessite beaucoup de données pour s'entraîner et prédire correctement

Pour ma problématique, un modèle de régression logistique surpasse un modèle de réseau de neurones à convolution