Modélisation d'Aptamères d'ADN IFT 3710

Dereck Piché, Jonas Gabirot, Guillermo Martinez

Avril 2023

Université de Montréal

Contexte

ADN-Aptamères

Propriétés fondamentales

1. Phénotype: affinité, se plie avec une fonction définie
2. Génotype (séquence): réplicable IN VITRO (SELEX)

Biosenseur

1. Structure chimique simple: 20-100 bases (insertion)
2. Liaison: changement conformationnel (signal)
3. Stabilité: supérieure aux ARN-aptamères

Minimum free energy (MFE) structure

Nanotechnologie et Computation

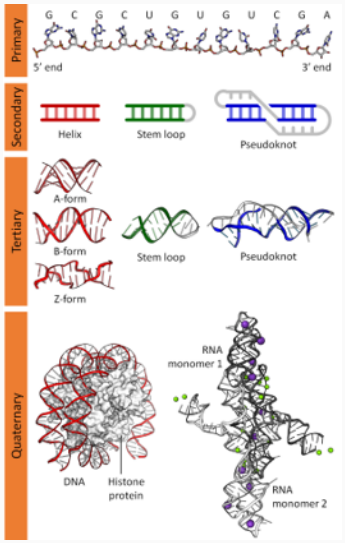
1. Stockage d'information: 2x supérieure au stockage binaire, pendant des milliers d'années
2. Qualité de stockage: dépend de la stabilité de la séquence.

Structure secondaire de l'ADN

1. Deux séquences (1D) qui se lient en une hélice (2D)
2. Stabilité: du pairage entre les bases des deux séquences, et donc de la taille et de l'arrangement des séquences

Minimum free energy (MFE) structure

Figure 1: Structures de l'ADN



Méthode NUPACK: estime MFE de la structure secondaire en fonction d'une analyse thermodynamique, Nearest-neighbor paramètres empiriques (Zadeh et al. 2010).

Objectif

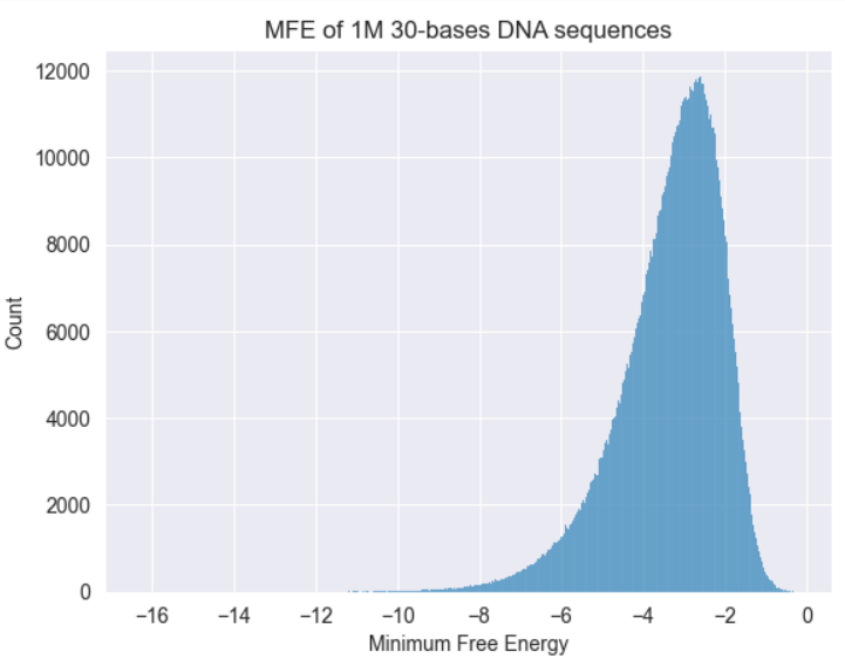
L'estimation du MFE d'une séquence d'ADN se complexifie en fonction de sa longeur. Cependant, les méthodes thermodynamiques comme celle de NUPACK sont rapides et relativement précises (r=70-80)

Objectif de recherche: entraîner des modèles d'apprentissage machine à prédire le MFE des structures secondaire de séquences d'ADN de 30 bases.

Retombées: L'apprentissage de représentations pour la fonction du MFE (étant donné des entrées séquentielles) peuvent réduire l'espace de recherche de réseaux génératifs, tels que les GFlowNets, pour générer des séquences d'ADN stables en minimisant la fonction apprise du MFE.

Visualisation données

Figure 2: Distribution des MFEs dans notre dataset



Baseline: MLP

MLP

Comme baseline, nous avons choisi un réseau MLP standard. Les MLPs ne font aucune supposition sur la structure de la séquence, ce qui permet de comparer avec d'autres modèles qui se basent sur des hypothèses différentes. Ils sont aussi faciles à implémenter et entraîner.

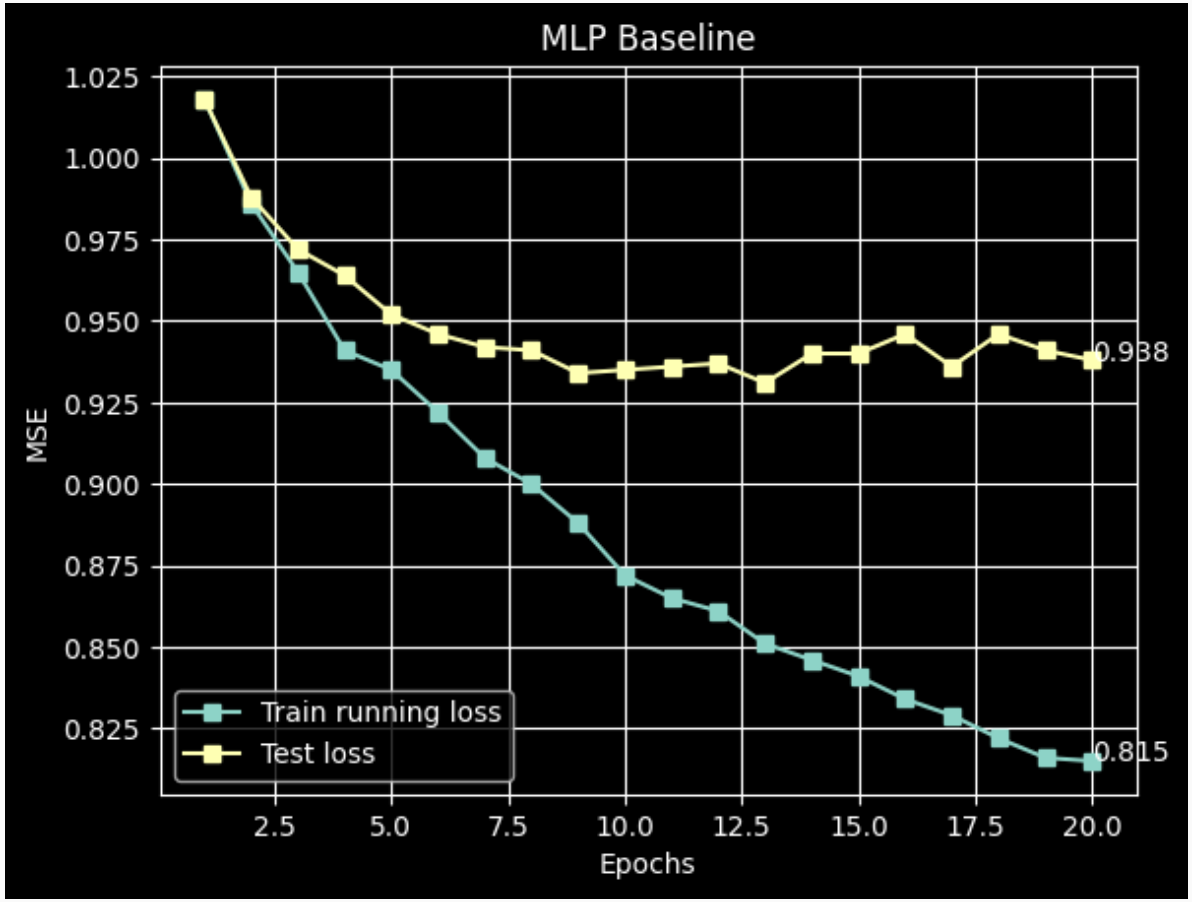
$$Seq \in R^{120} \xrightarrow{Lin.(\times 20)} \xrightarrow{ReLU(\times 20)} \xrightarrow{Lin.} Energie \in R$$

Hyperparamètres

1. Taux d'apprentissage: 0.0003
2. Nombre de paramètres: 1 005 241

Résultats du MLP

Figure 3: Précision du MLP selon MSE



Long Short Term Memory (LSTM)

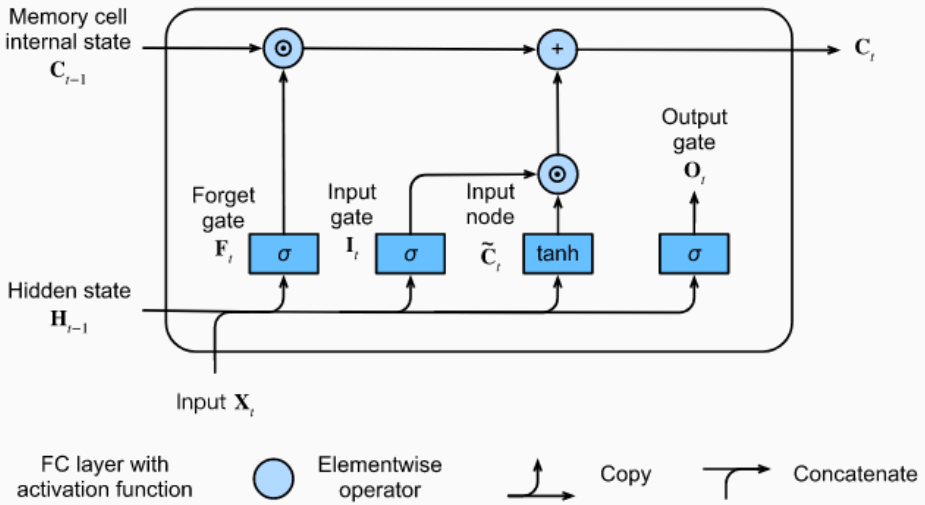
LSTM

Le modèle LSTM (Long Short-Term Memory) est une variante des réseaux de neurones récurrents (RNN) qui permet de mieux gérer les problèmes liés à la mémoire à long terme. Il utilise des portes, qui sont des fonctions non linéaires, pour contrôler l'information qui entre ou sort de la mémoire.

Portes

Il y a trois types de portes : la porte d'oubli (forget gate) permet au modèle de décider ce qu'il faut oublier de la mémoire à long terme, la porte d'entrée (input gate) permet d'ajouter de nouvelles informations à la mémoire à long terme et la porte de sortie (output gate) permet de déterminer quelle information doit être renvoyée en sortie.

Figure 4: Architecture LSTM



Architecture du LSTM

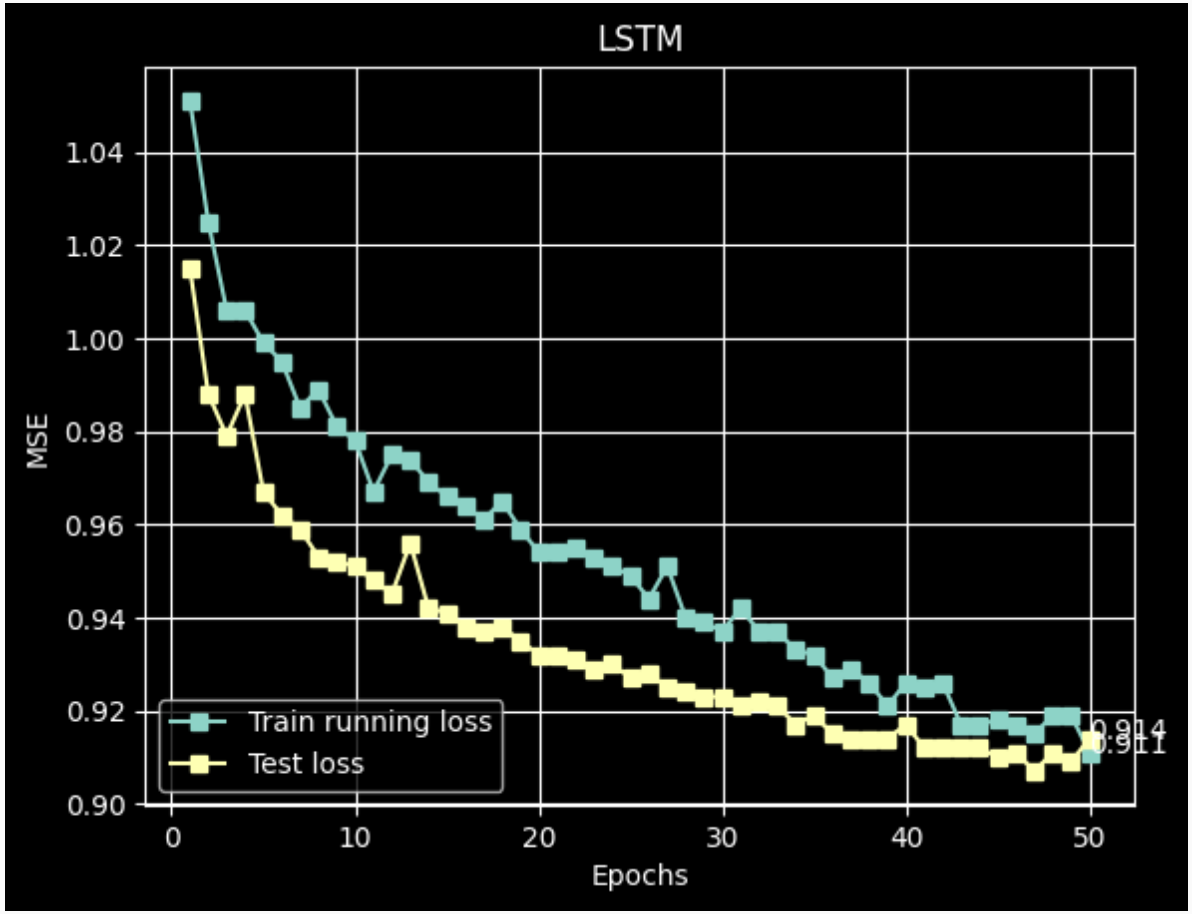
$$Seq \in R^{120} \xrightarrow{LSTM1} \xrightarrow{LSTM2} \xrightarrow{Lin.} Energie \in R$$

Hyperparamètres

1. Taux d'apprentissage: 0.0003
2. Nombre de paramètres: 1 008 991
3. Dropout: 0.6

Résultats du LSTM

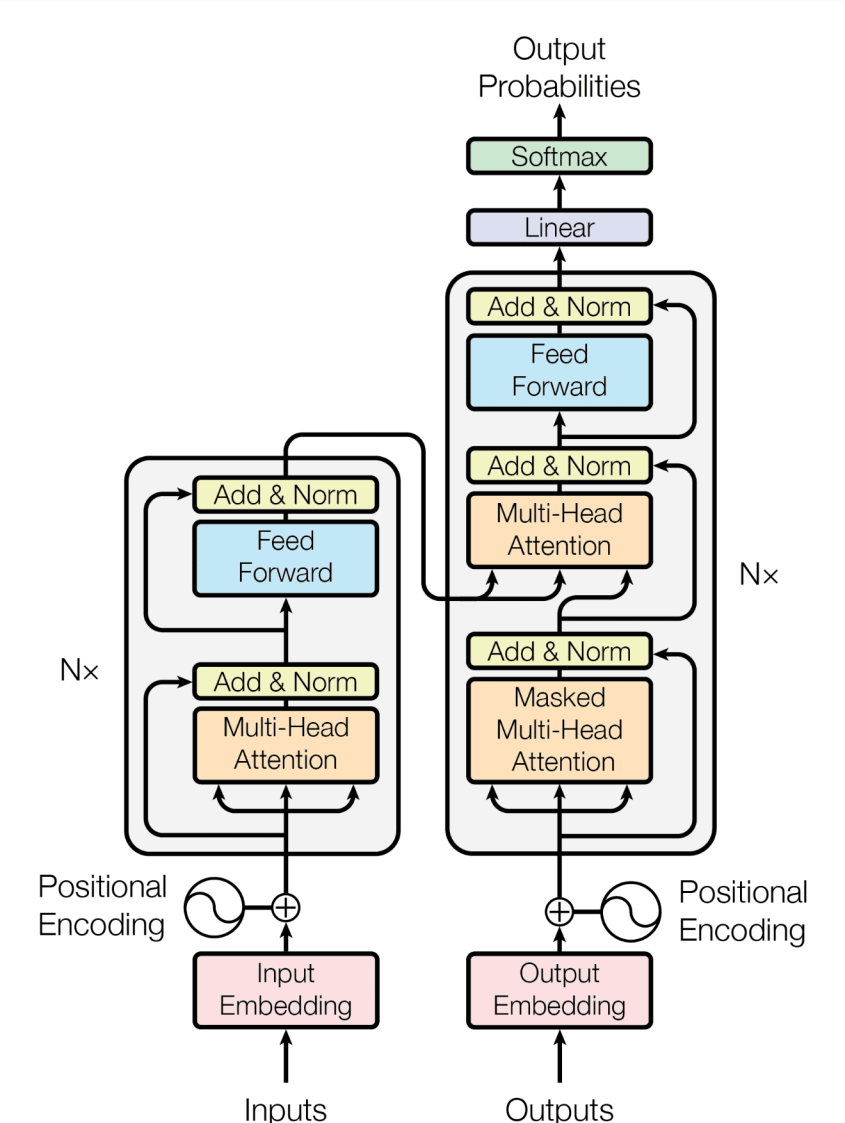
Figure 5: Précision du LSTM selon MSE



Transformeurs

Architecture du transformeur

Figure 6: Transformer de traduction dans AIAYN



Vectorisation

Avant d'entrer dans le coeur du transformeur (les couches de codage), les séquences sont vectorisés. Le mécanisme de vectorisation est appris. Ensuite, il est pratique courante de sommer un vecteur positionnel sur la vectorisation de la séquence. Ce vecteur, tel que son nom l'indique, contient de l'information sur la position des jetons de la séquence. Il existe plusieurs méthodes pour générer ce vecteur positionnel. Nous avons utilisé la méthode proposée dans le papier AIAYN.

Attention

Un système d'attention retourne une somme des vecteurs *Valeurs* selon une mesure de similarité entre les vecteurs *Requêtes* et les vecteurs *Clés*. Cette mesure de similarité est appelée l'attention.

Système d'attention

$\sum_j a(q, k_j) v_j$

Les transformeurs utilisent un système d'auto-attention, qui vise à produire une séquence transformée dans laquelle chaque jeton est codé en effectuant une somme pondérée des autres jetons présents dans sa séquence (l'incluant).

Jeton codé

$t_i' = \sum_j a(W^q t_i, W^k t_j) W^v t_j$

Attention

Dans les transformeurs, on utilise une attention spéciale :

Attention au produit scalaire mis à l'échelle

$$a(q, k) = softmax(\frac{q^T k}{\sqrt{d}})$$

Pourquoi le produit scalaire? On sait déjà que le produit scalaire peut être vu comme une mesure de la similarité entre les deux vecteurs. Cela permet beaucoup de liberté au transformeur, qui obtient les vecteurs clés et les vecteurs requêtes en fonction des vecteurs jetons avant d'effectuer le produit scalaire. On le laisse donc créer sa propre mesure de distance, en quelque sorte.

Pourquoi le softmax? Pour que les coéfficients somment à 1. (*expliquer l'intuition)

Pourquoi la division par la racine de la taille? Pour éviter des erreurs numériques. On épargne les détails.

Couche de codage

Une tête d'attention retourne donc une séquence de la même taille que son entrée (sauf exceptions). Une couche de codage contient plusieurs têtes d'attention! Les sorties de ces têtes d'attention sont concaténées dans *un gros vecteur* et on applique une transformation linéaire (avec activation ReLU pour obtenir de l'expressivité non linéaire) à ce vecteur pour obtenir une sortie de dimension arbitraire.

Architecture de notre transformeur de régression

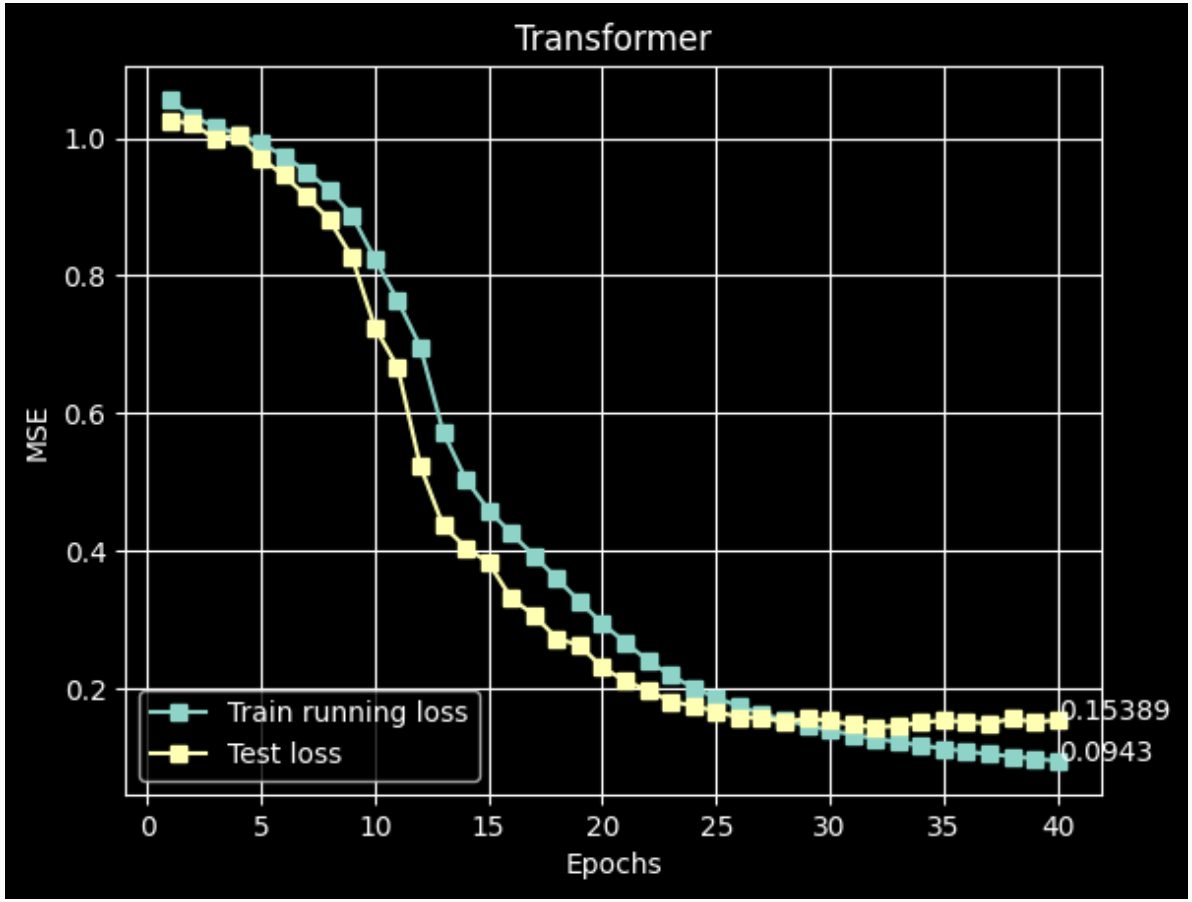
$$Seq \in R^{30} \xrightarrow{Vect.(Dim=16)} \xrightarrow{+Posit.} \xrightarrow{Cod.(\times 3)} \xrightarrow{Lin.} Energie \in R$$

Hyperparamètres

1. Taux d'apprentissage: 10^{-4}
2. Dimension de vectorisation: 16
3. Nombre de têtes: 8
4. Nombre de paramètres: 1 786 753
5. Codage positionnel: sincos (AIAYN)

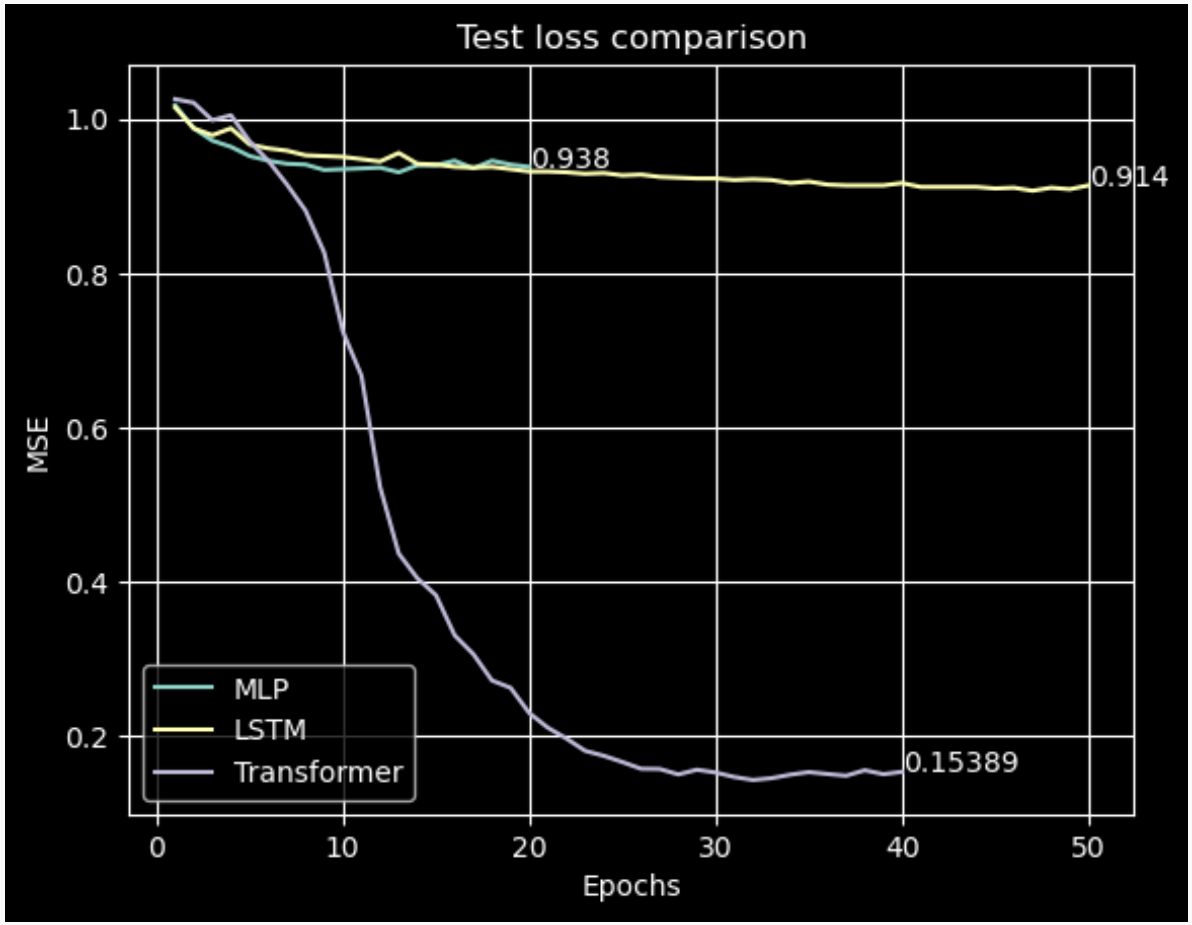
Résultats du transformeur

Figure 7: Précision du transformeur selon MSE



Comparaison des résultats

Figure 8: MSE des 3 modèles

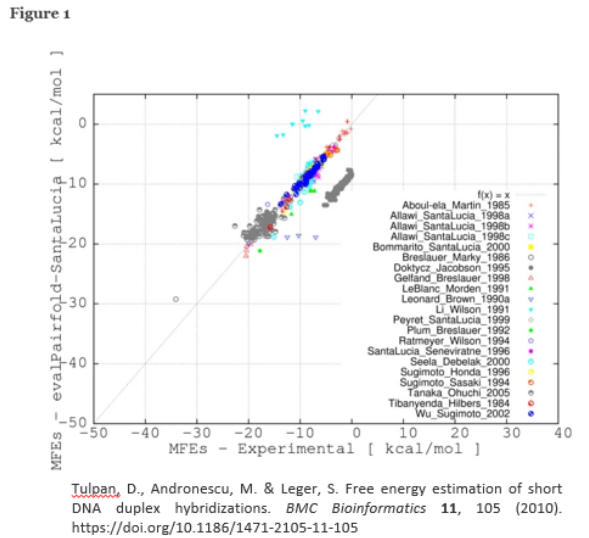


Conclusion

Contraste MFE expérimental/simulation

Technique d'estimation thermodynamique du MFE similaire à celle de NUPACK: r=75 avec MFE expérimentale.

Figure 9: Distribution des MFEs dans notre dataset



Tulpan, D., Andronescu, M. & Leger, S. Free energy estimation of short DNA duplex hybridizations. *BMC Bioinformatics* **11**, 105 (2010). https://doi.org/10.1186/1471-2105-11-105

Suggestions d'étapes à suivre

1. Tester la validité prédictive du Transformer sur des séquences ADN dont le MFE a été établi empiriquement.
2. Augmenter la taille du dataset à 10-100 million de séquences pour inclure des plus fortes minimas en termes de MFE. Puis, tester la validité prédictive du transformer sur ces minimas. Re-entraîner au besoin.
3. Entraîner des GFlowNets à générer des séquences d'ADN qui minimisent le MFE, pour aller au-delà de l'aléatoire.
4. Compléter le protocole End-to-End-DNA (SELEX) pour transformer nos séquences d'ADN les plus stables en aptamères avec des affinités spécifiques.