Stanford Ribonanza RNA Folding

Prédiction des structures locales de l'ARN à partir des séquences nucléotidiques par apprentissage profond

UE Intelligence artificielle avancée - M2BI

Dahirou SAM, Bilal DELIKAYA, Karim BOUCHAARA, Anaïs DELASSUS

Année 2025-2026

Introduction :

L'acide ribonucléique (ARN) n'est pas seulement un messager : ses structures secondaires et tertiaires gouvernent directement de nombreuses fonctions cellulaires : catalyse, reconnaissance moléculaire, régulation de l'expression et conditionnent l'impact fonctionnel de mutations. Maîtriser le lien séquence et structure est donc un prérequis scientifique et technologique pour la conception rationnelle de médicaments à base d'ARN, de vaccins ARNm, et d'outils de biologie synthétique.

La détermination expérimentale de structures d'ARN par cristallographie, RMN ou Cryo-EM reste coûteuse, parcellaire et parfois inadaptée aux molécules flexibles ou à l'étude en conditions proches du physiologique. Les méthodes expérimentales de lecture indirecte du repliement, comme la cartographie chimique (DMS (Diméthyle Sulfate), 2A3(2-Aminopyridine-3-carboxylic acid imidazolide)) couplée au Mutational Profiling (MaP),

fournissent des profils positionnels de réactivité chimique qui reflètent l'accessibilité et l'appariement des nucléotides. Ces profils sont des signatures structurales riches : prédire correctement ces mesures à partir d'une simple séquence reviendrait à posséder une représentation implicite, et utile, du repliement moléculaire. Toutefois, l'apprentissage automatique sur ces données bute sur des défis concrets tels que la variabilité expérimentale, le bruit, problème de positions non sondées en bordure, redondances d'expérience etc… qui exigent des choix de prétraitement et d'évaluation scrupuleux.

La compétition Ribonanza RNA Folding, organisée sur la plateforme Kaggle par le laboratoire de Rhiju Das à l'Université de Stanford, met à disposition une vaste collection de profils MaP diversifiés. Elle offre un cadre d'évaluation rigoureux et réaliste pour développer des modèles capables de généraliser à des séquences nouvelles. Réussir à prédire les profils DMS et 2A3 signifie non seulement améliorer la compréhension fondamentale du repliement, mais aussi fournir un outil prédictif immédiatement utile pour la recherche et la conception d'ARN thérapeutiques.

Ce projet vise précisément à tirer parti de ces données pour construire et valider des modèles d'apprentissage profond capables d'estimer, position par position, la réactivité chimique d'un ARN donné. L'objectif est pragmatique et ambitieux : fournir une méthode reproductible et évaluée rigoureusement qui rapproche la prédiction de structure d'une application concrète en biologie et en médecine.

Matériels et Méthodes :

Données expérimentales :

Les jeux de données utilisés proviennent d'expériences de cartographie chimique appliquées à des molécules d'ARN et fournies dans le cadre de la compétition Ribonanza. Pour chaque échantillon, la base de données contient la séquence nucléotidique (A, C, G, U), des profils positionnels de réactivité chimique obtenus avec deux réactifs distincts (DMS et 2A3), des estimations d'erreur expérimentale associées à ces mesures, ainsi que des métadonnées qualitatives et quantitatives telles que le nombre de lectures (« reads »), le rapport signal/bruit (« signal_to_noise ») et un indicateur binaire de qualité global (« SN_filter »). Les profils de réactivité sont issus d'un pipeline MaP : après traitement chimique et rétrotranscription, les mutations induites sont détectées par séquençage à haut débit et converties en valeurs continues de réactivité pour chaque position de la séquence. Du fait des limites expérimentales, plusieurs positions en tête et en queue de séquence ne sont pas sondées et sont représentées par des valeurs manquantes (NaN).

Exploration des données et filtrage :

Exploration :

Un premier jeu de données nous a été mis à disposition pour ce projet. Celui-ci est composé au total de 1 643 680 observations dont 806 578 séquences d'ARN uniques. Il devrait à priori servir d'entraînement pour les modèles. Un second jeu de données composé de 1 031 888 séquences qui devrait servir pour le test a été aussi mis à disposition.
L'exploration initiale a mis en évidence des différences marquées entre les distributions de longueur des séquences dans les ensembles d'apprentissage et de test : les séquences d'entraînement varient principalement entre 115 et 206 nucléotides, avec une forte concentration autour de 175 nt, alors que les séquences du test s'étendent jusqu'à 457 nt.

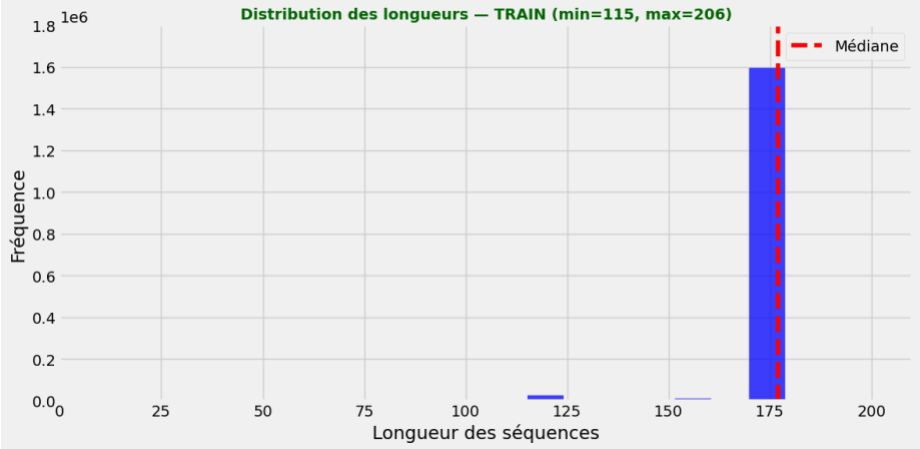

Figure 1A: Distribution des longueurs des séquences d'ARN dans le jeu d'entraînement

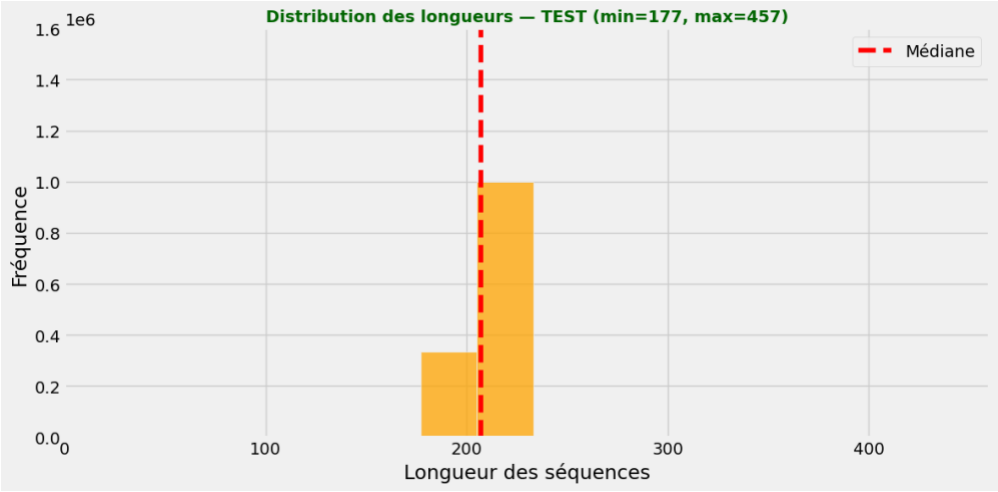

Figure 1B: Distribution des longueurs des séquences d'ARN dans le jeu de test

Cette dissymétrie impose un soin particulier lors du prétraitement pour garantir la capacité de généralisation des modèles aux séquences plus longues. Afin d'assurer la qualité des profils retenus pour l'apprentissage, nous avons appliqué un filtrage en plusieurs étapes.

Filtrage :

Différents filtres ont été appliqués au jeu de données d'entrainements. D'abord, seules les observations avec l'indicateur SN_filter == 1, correspondant à un rapport signal/bruit supérieur à 1.0 et à un nombre de lectures supérieur à 100 (reads_threshold = 100, SNR_threshold = 1.0) ont été conservées en retirant les profils présentant simultanément un faible nombre de lectures et un faible SNR. Ce qui nous a permis de passer à 437 912 séquences, soit 26.6% des données initiales.

Par ailleurs, pour chaque couple (séquence, type d'expérience) nous avons supprimé les réplicas en ne conservant que la mesure associée au signal_to_noise maximal, ce qui privilégie la réplication la plus fiable pour une même séquence. On passe alors à 335956 séquences, soit 20.4% avec 167 978 séquences uniques. C'est sur ce jeu données de 167 978 séquences uniques que nous travaillons pour la suite.

Ainsi, conformément aux bonnes pratiques d'évaluation, le découpage en ensembles d'entraînement, validation et test a été effectué sur séquences uniques afin d'empêcher toute fuite d'information entre splits ; la partition retenue est 70 % / 18 % / 12 % (train / val / test).

Prétraitement et encodage des séquences :

Encoding:

Pour rendre les séquences utilisables par des modèles d'apprentissage automatique, chaque nucléotide a été encodé par un vecteur One-Hot de dimension quatre (A, C, G, U).One-hot encoding : mapping {A, C, G, U} en vecteurs de dimension 4 : A [1,0,0,0], C [0,1,0,0], G [0,0,1,0], U [0,0,0,1]. Chaque séquence est donc initialement représentée par une matrice (L, 4) où L est la longueur de la séquence. Cette représentation simple mais neutre préserve la nature discrète des lettres tout en étant directement exploitable par des couches linéaires, récurrentes ou d'attention. Les profils de réactivité associés à DMS et 2A3 ont été conservés comme cibles à deux canaux distincts.

Normalisation :

Les valeurs expérimentales de réactivité présentent des distributions asymétriques et contiennent des outliers : pour stabiliser l'entraînement et limiter l'impact des mesures extrêmes, les réactivités ont été normalisées par un z-score robuste. Concrètement, pour chaque réactif nous avons calculé la médiane mm sur l'ensemble des valeurs observées (hors NaN) et la médiane des écarts absolus (MAD), puis appliqué la transformation $(x-m)/(1.4826\times MAD)$. Le coefficient 1.4826 est le facteur usuel qui rend la MAD comparable à l'écart-type pour une distribution gaussienne, ce qui facilite l'interprétation et la convergence numérique.

Masking et Padding :

Les valeurs manquantes (positions non sondées) ont été laissées en NaN au niveau du jeu de données et gérées ensuite par masquage pendant le calcul de la perte.
La variabilité de longueur des séquences a conduit à appliquer un padding uniforme à la longueur maximale observée dans le jeu de test, soit 457 positions. Chaque schéma d'entrée est donc représenté par une matrice d'entrée XX de shape (457, 4) et chaque cible YY par une matrice de shape (457, 2). Les positions au-delà de la longueur réelle sont remplies par des zéros. Pour éviter que le modèle n'apprenne sur ces artefacts de padding, un masque binaire a été construit pour chaque instance : ce masque vaut 1 pour les positions réellement mesurées et 0 pour les positions de padding et les positions non-probed; il est appliqué à l'avance et pendant l'agrégation des pertes. Lors de l'entraînement, la fonction de perte masquée est calculée ainsi : la somme des carrés des erreurs est pondérée par le masque, puis normalisée par le nombre de positions valides (avec une petite constante ajoutée pour prévenir la division par zéro). Cette stratégie garantit que seules les contributions expérimentales réelles influent sur la mise à jour des paramètres.

Modèles, stratégie d'entraînement et mesures d'évaluation :

Quatre familles d'architectures ont été implémentées et comparées pour leur capacité à prédire les profils positionnels : un LSTM simple, un modèle récurrent bidirectionnel (BiLSTM), un modèle combinant LSTM et convolution 1D, et une architecture basée sur des blocs Transformer adaptés aux features continues.

Le BiLSTM utilise deux couches LSTM empilées, la première bidirectionnelle avec 128 unités et la seconde avec 64 unités, complétées par des normalisations de couche (LayerNorm) et des couches de dropout (taux 0,3) pour limiter le sur-apprentissage ; la sortie est temporelle et produit deux valeurs par position par l'intermédiaire d'une couche dense appliquée en mode « time-distributed ».

Le modèle LSTM+Convolution combine deux couches LSTM (64 puis 32 unités) suivies d'un ou plusieurs filtres convolutionnels 1D pour capter des motifs locaux récurrents ; il utilise un dropout réduit (0,1) et un mécanisme d'early stopping.

Le Transformer a été conçu pour accepter en entrée des features continues (one-hot projeté linéairement) via un bloc d'embedding continu positionnel ; il empile deux blocs Transformer (embed_dim = 64, 4 têtes d'attention, ff_dim = 128), applique un pooling global puis projette vers les deux cibles. Le choix de la projection linéaire en entrée (ContinuousTokenEmbedding) évite l'utilisation d'indices entiers dans nn.Embedding et résout les incompatibilités dimensionnelles entre embedding de position et features continues.
Les hyperparamètres partagés comprennent un batch_size de 128 et l'optimiseur Adam avec un taux d'apprentissage initial de 1×10^{-3}. Les modèles ont été entraînés typiquement sur 40–50 époques, avec early stopping basé sur la perte de validation (patience = 5) et sauvegarde du meilleur checkpoint.

La fonction de perte principale est la MSE masquée ; les métriques de suivi comprennent la MSE, la MAE et la RMSE, toutes calculées en appliquant le même masque que la perte afin d'évaluer exclusivement les positions observées.

Les expérimentations ont été réalisées sous PyTorch, avec exploitation de l'accélération matérielle Apple MPS quand elle était disponible ; des seeds et random_state ont été fixés pour garantir la reproductibilité et les DataLoaders ont été configurés avec un nombre de workers adapté au système pour éviter une surcharge mémoire sur macOS.

Enfin, l'évaluation finale repose non seulement sur les métriques numériques agrégées, mais aussi sur des diagnostics visuels : courbes d'apprentissage (loss / MAE par époque), heatmaps comparatives des profils réels et prédits sur sous-ensembles représentatifs, tracés positionnels de la prédiction vs la vérité expérimentale et distributions d'erreurs afin d'identifier d'éventuels biais positionnels ou dépendances non modélisées.

Résultats :

Nous avons évalué quatre architectures pour la prédiction, position par position, des réactivités DMS et 2A3 à partir des séquences d'ARN encodées en One-Hot (457 × 4). Toutes les expériences ont été réalisées avec le même pipeline de données, et un clip des prédictions dans [0,1]. Le découpage en ensembles d'entraînement, de validation et de test est identique pour tous les modèles, ce qui permet de comparer les performances uniquement sur la base des différences architecturales. Les performances reportées sont des MAE masquées en validation et en test.

LSTM simple

Le premier modèle testé correspond à une architecture LSTM simple, composée d'une seule couche LSTM suivie d'une couche dense linéaire à deux neurones appliqués à chaque position de la séquence. Ce modèle sert de référence de base pour évaluer les améliorations apportées par des architectures plus complexes.

La courbe de MAE (Figure 4) montre une diminution rapide et régulière de l'erreur dès les premières époques, indiquant que le modèle apprend efficacement les motifs principaux présents dans les données. La courbe de validation suit étroitement celle de l'entraînement, ce qui traduit une bonne stabilité de l'apprentissage et une absence notable de surapprentissage. La convergence est atteinte aux alentours de 30 à 35 époques.

Bien que le modèle parvienne à capturer la structure globale des séquences courtes et moyennes, sa capacité à généraliser aux séquences plus longues reste limitée. En effet, sans

mécanisme de bidirectionnalité ni de prise en compte explicite des motifs locaux, cette architecture basique peine à modéliser les dépendances complexes dans les signaux de réactivité.

En résumé, cette baseline fournit un point de comparaison stable et simple, sur lequel les améliorations des architectures plus complexes pourront être clairement mesurées.

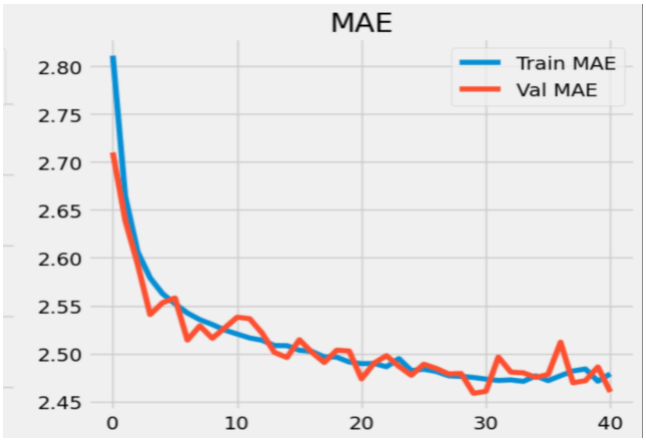

Figure 2 : Courbe d'évolution de la MAE pour le modèle LSTM simple

Bi-LSTM

Ce deuxième modèle repose sur une architecture plus complexe que la baseline. Il est constitué d'une couche BiLSTM suivie d'une LSTM classique, d'un dropout et de deux couches denses finales. L'ajout de la bidirectionnalité permet au réseau de prendre en compte les dépendances dans les deux sens le long de la séquence, enrichissant ainsi la représentation contextuelle de chaque position.

La courbe de MAE (Figure 3) montre une convergence nette et stable, avec une diminution progressive de la MAE à la fois pour l'ensemble d'entraînement et de validation. Contrairement au modèle LSTM simple, la MAE de validation est systématiquement plus basse que celle de l'entraînement. Ce phénomène peut s'expliquer par la présence de dropout, qui agit comme une régularisation efficace, ainsi que par une meilleure généralisation apportée par la bidirectionnalité. La courbe de validation suit une trajectoire plus ou moins lisse, ce qui traduit un apprentissage stable et une bonne capacité à s'adapter aux données sans surapprentissage marqué.

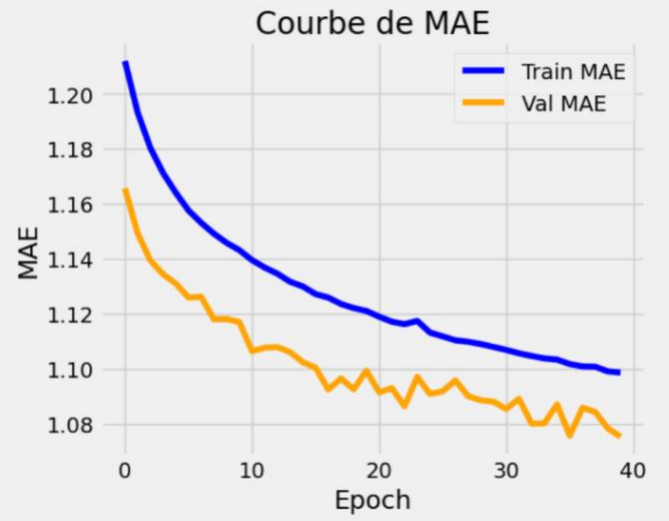

Figure 3: Courbe d'évolution de la MAE pour le modèle BiLSTM

La comparaison entre valeurs réelles et valeurs prédites en figure 6 illustre la capacité du modèle à capturer fidèlement la structure globale et locale des profils de réactivité. Les pics principaux observés dans le signal réel sont correctement reproduits dans les prédictions. Le modèle parvient également à suivre les fluctuations fines, notamment dans les régions à forte variabilité. Quelques petites divergences apparaissent pour certains pics très fins ou bruités, mais l'alignement global est nettement meilleur que pour le LSTM simple.

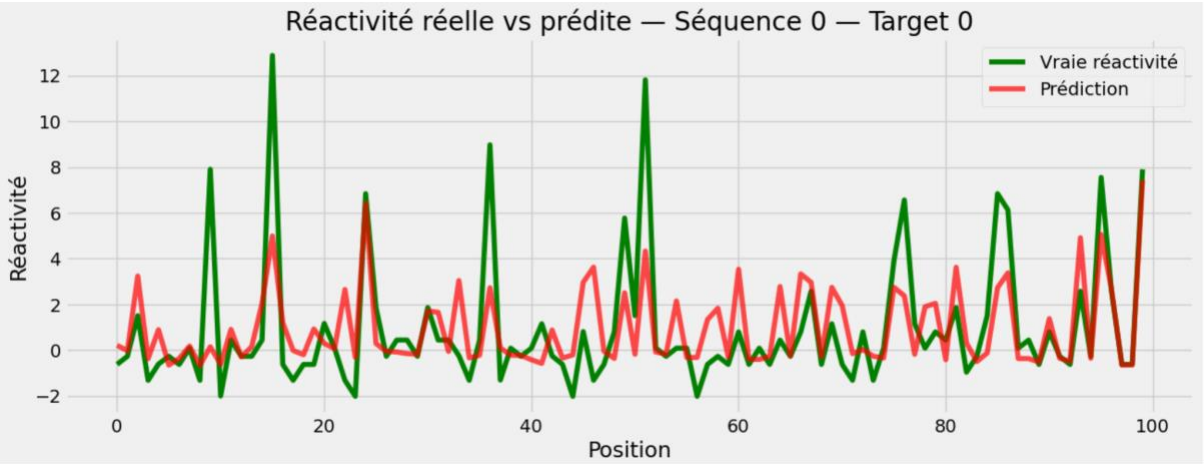

Figure 4: Comparaison entre les réactivités réelles et prédites sur une séquence représentative pour le modèle Bi-LSTM avec DMS

L'amélioration obtenue s'explique par la capacité du BiLSTM à intégrer des informations venant de l'amont et de l'aval de chaque position, permettant ainsi de mieux situer chaque nucléotide dans son contexte global. Cela se traduit par une réduction significative de la MAE, mais également par des prédictions plus cohérentes et plus précises au niveau local.

En résumé, l'introduction de la bidirectionnalité dans l'architecture LSTM permet d'obtenir un gain net en performance, aussi bien en termes de stabilité de l'entraînement que de fidélité

de la prédiction des profils de réactivité.

LSTM et Couches de Convolution

Le troisième modèle combine une architecture BiLSTM avec des convolutions 1D dilatées. Concrètement, il applique une couche BiLSTM suivie d'une LSTM classique, puis de trois couches convolutives dilatées (dilations 1, 2 et 4) avant les couches denses finales. Cette architecture a été pensée pour capturer à la fois le contexte global via les LSTM et les motifs locaux multi-échelles grâce aux convolutions, tout en restant moins coûteuse en calcul qu'un Transformer.

La courbe de MAE (Figure 5) montre une diminution progressive de l'erreur pour les ensembles d'entraînement et de validation. La MAE de validation reste très proche de celle de l'entraînement sur l'ensemble des époques. On note une certaine irrégularité de la baisse de l'erreur avec des pic notables entre 10 et 15 epochs. La convergence est atteinte aux alentours de 40 à 50 époques, en cohérence avec le seuil de patience de l'early stopping.

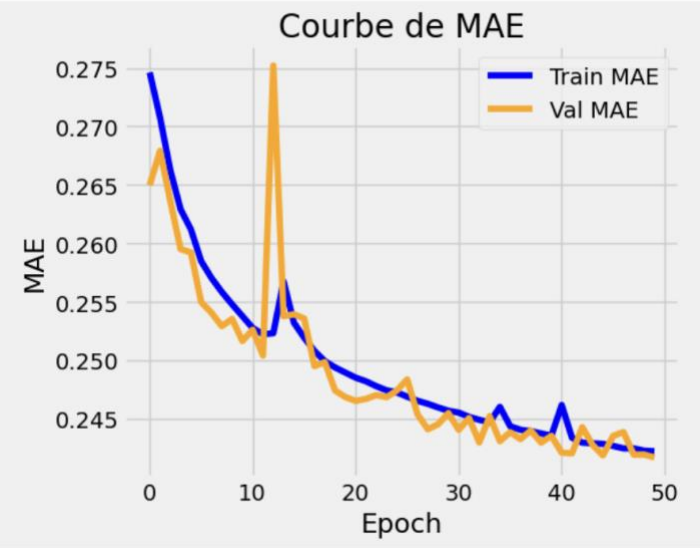

Figure 5: Courbe d'évolution de la MAE pour le modèle LSTM + Conv1D dilatées.

La figure 6 compare les valeurs de réactivité réelles et prédites sur une séquence représentative. On observe que le modèle parvient à restituer généralement les pics majeurs de réactivité, notamment aux positions où les signaux sont les plus intenses. Les fluctuations locales sont également bien suivies, ce qui témoigne de la capacité des convolutions dilatées à capter des motifs à différentes échelles. Ce modèle arrive bien à prédire la réactivité pour cette séquence aussi bien que le BiLSTM.

Cependant, on note une légère sous-estimation de l'amplitude de certains pics très fins et bruités. Ce phénomène, déjà observé dans des travaux similaires, reflète une tendance du

modèle à lisser les extrêmes pour mieux généraliser. Cela n'empêche toutefois pas le modèle de bien reproduire la structure globale et les grandes tendances des profils de réactivité.

Cette combinaison LSTM + Conv1D dilatées permet donc de tirer parti des deux approches : la modélisation séquentielle globale et l'extraction locale de motifs. Elle améliore nettement la qualité des prédictions tout en maintenant un coût d'entraînement raisonnable.

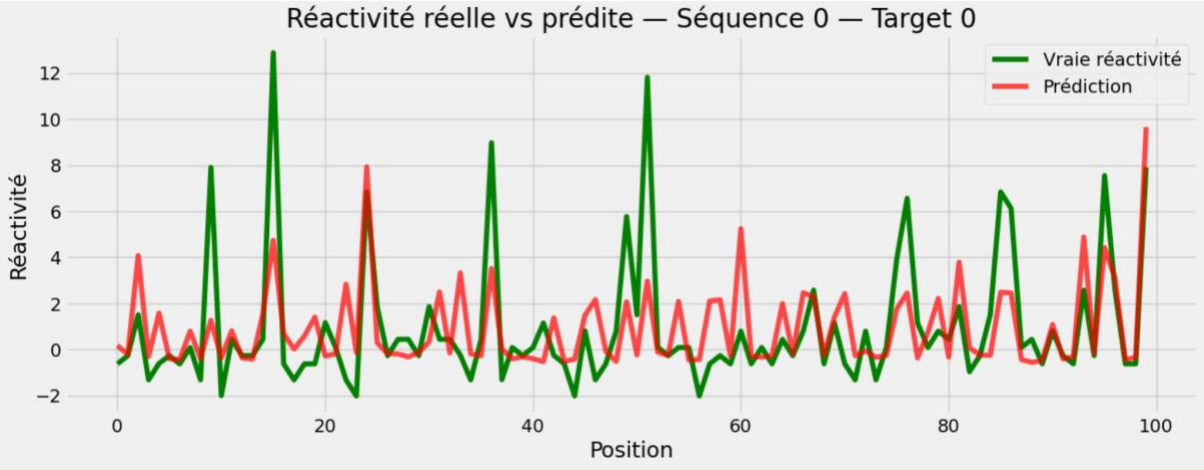

Figure 6: Réactivité réelle et prédite sur une séquence représentative avec DMS

Transformer

Le dernier modèle évalué repose sur une architecture Transformer, composée d'un embedding combinant token et position, suivi de deux blocs Transformer classiques (Multi-Head Self Attention + Feed Forward), puis d'une projection linéaire à deux neurones par position. Ce type de modèle est particulièrement adapté aux séquences longues, car il permet de relier directement des positions distantes, contrairement aux architectures séquentielles qui propagent l'information de manière locale.

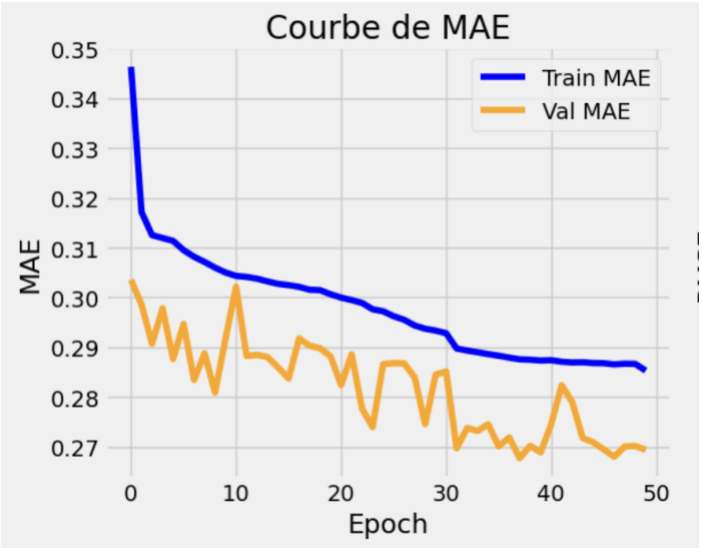

Figure 7: Courbe d'évolution de la MAE pour le modèle Transformer

La courbe de MAE (Figure 7) montre une convergence rapide et stable, avec une réduction marquée de la MAE dès les premières époques. La MAE de validation est systématiquement inférieure à celle de l'entraînement, comme pour le BiLSTM mais reste très instable. Cette différence pourrait s'expliquer par la taille des données par rapport à la complexité du modèle.

La figure 8 présente la relation entre les valeurs de réactivité prédites et les valeurs réelles pour les deux cibles (Target 0 = DMS, Target 1 = 2A3). Chaque point représente une position dans une séquence d'ARN. La diagonale rouge correspond à la parfaite corrélation (y = x). On observe une forte densité de points concentrés le long de cette diagonale, ce qui indique une bonne calibration globale du modèle. La majorité des prédictions sont proches des valeurs réelles, même pour des amplitudes élevées.

On remarque toutefois une dispersion légèrement plus marquée pour les valeurs extrêmes (pics de forte réactivité), en particulier pour la Target 0 (DSM). Cela traduit une tendance du modèle à sous-estimer légèrement les valeurs maximales, ce qui est cohérent avec les observations faites pour le LSTM+Conv. Néanmoins, la structure en attention du Transformer permet de mieux reproduire la dynamique globale des signaux, en particulier sur les séquences longues.

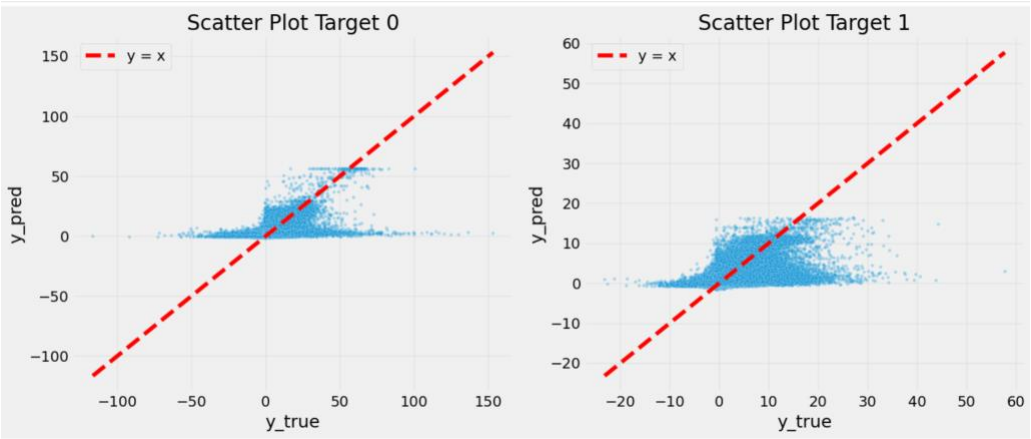

Figure 8 : Comparaison entre la réactivité réelle et prédite pour une séquence représentative.

La figure 9 illustre cette capacité sur une séquence donnée. Les pics de réactivité sont globalement bien alignés, y compris dans les régions éloignées du début de la séquence, grâce à la capacité du modèle à modéliser efficacement les dépendances à longue portée. Les fluctuations locales sont également bien reproduites, ce qui traduit une modélisation fine des motifs structuraux.

Sur cette séquence particulière, le modèle avec transformer donne des résultats similaires avec le BiLSTM et le LSTM plus Convolution.

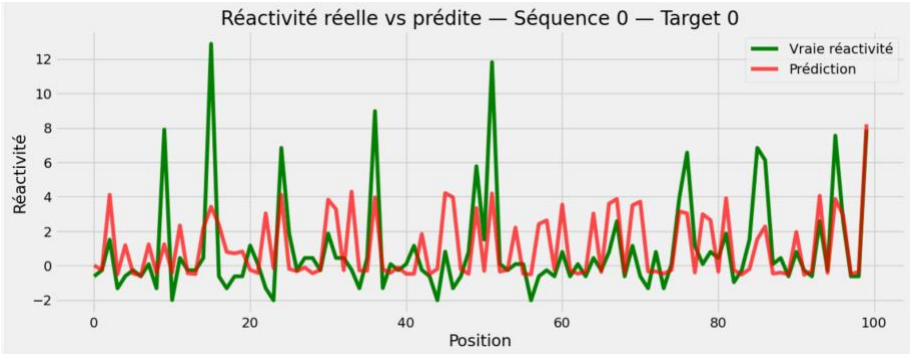

Figure 9: Comparaison entre les réactivités réelles et prédites sur une séquence représentative pour le modèle Transformer avec DMS

Discussion :

Il est important de souligner que les modèles comparés ne sont pas strictement comparables entre eux. Plusieurs facteurs peuvent influencer les résultats et limiter la validité d'une comparaison directe. Les fonctions de perte utilisées peuvent varier légèrement selon les architectures, certaines implémentant par exemple un masking spécifique ou une réduction différente. De même, les métriques calculées (MSE, MAE, RMSE) peuvent être implémentées de manière distincte, notamment pour gérer les séquences de longueur variable ou les valeurs manquantes. Par ailleurs, les paramètres d'entraînement tels que le learning rate, le dropout, la taille des batches, le scheduler ou la patience de l'early stopping ne sont pas forcément identiques entre les modèles, ce qui peut introduire un biais sur la performance finale. Enfin, la complexité accrue de certains modèles peut les rendre plus sensibles au bruit des données ou au sur-apprentissage, en particulier pour des séquences longues. Il convient également de noter que les ressources de calcul étaient limitées, ce qui a restreint la possibilité de réaliser plusieurs runs ou d'explorer davantage les hyperparamètres, limitant ainsi la capacité à apporter certaines corrections et à valider pleinement la robustesse des modèles.

Conclusion :

Ce projet visait à prédire les profils de réactivité DMS et 2A3 à partir de séquences d'ARN, en utilisant des modèles d'apprentissage profond adaptés aux séquences de longueur variable. Un pipeline complet : nettoyage, encodage, normalisation, padding et masquage a été mis en place pour stabiliser l'apprentissage.

Quatre architectures ont été comparées : LSTM, BiLSTM, LSTM + convolutions dilatées et Transformer. Les résultats montrent que la complexité du modèle améliore généralement la précision, mais avec certaines limites : le Transformer, malgré sa capacité à capturer les dépendances longues, n'a pas toujours surpassé de manière significative les modèles moins complexes comme le LSTM + convolutions. Ce phénomène peut s'expliquer par la taille finie des données, le bruit présent dans les mesures expérimentales et la difficulté d'entraîner des architectures très profondes sur ce type de séquences.

Le LSTM + convolutions reste le meilleur compromis entre performance et coût computationnel, tandis que le Transformer peut apporter un gain marginal de précision mais avec un entraînement plus coûteux et une sensibilité plus grande au surapprentissage.

Bibliographie :

1. Vicens, Q. et Kieft, J.S. (2022) « Thoughts on how to think (and talk) about RNA structure », Proceedings of the National Academy of Sciences, 119(17), p. e2112677119. Disponible sur: https://doi.org/10.1073/pnas.2112677119.

2. Deng, J. et al. (2023) « RNA structure determination: From 2D to 3D », Fundamental Research, 3(5), p. 727-737. Disponible sur: https://doi.org/10.1016/j.fmre.2023.06.001.

3. Deigan, K.E. et al. (2009) « Accurate SHAPE-directed RNA structure determination », Proceedings of the National Academy of Sciences, 106(1), p. 97-102. Disponible sur: https://doi.org/10.1073/pnas.0806929106.

4. Rouskin, S. et al. (2014) « Genome-wide probing of RNA structure reveals active unfolding of mRNA structures in vivo », Nature, 505(7485), p. 701-705. Disponible sur: https://doi.org/10.1038/nature12894.

5. Bliss, N., Bindewald, E. et Shapiro, B.A. (2020) « Predicting RNA SHAPE scores with deep learning », RNA Biology, 17(9), p. 1324-1330. Disponible sur: https://doi.org/10.1080/15476286.2020.1760534.

6. Wayment-Steele, H.K. et al. (2022) « RNA secondary structure packages evaluated and improved by high-throughput experiments », Nature Methods, 19(10), p. 1234-1242. Disponible sur: https://doi.org/10.1038/s41592-022-01605-0.

7. Hochreiter, S. et Schmidhuber, J. (1997) « Long Short-Term Memory », Neural Computation, 9(8), p. 1735-1780. Disponible sur: https://doi.org/10.1162/neco.1997.9.8.1735.