**OpenVaccine: COVID-19 mRNA Vaccine Degradation Prediction**



|  |  |
| --- | --- |
| **Trinôme** | Aude Sterlin  Nicolas Bisson  Sleheddine Kastalli |
| **Année universitaire** | 2020/2021 |

Introduction (contexte général)

La raison pour laquelle la stabilité de l'ARNm a un rapport avec les vaccins est que les vaccins à base d'ARNm sont une technologie émergente. Quelques entreprises disposent de vaccins à ARNm pour le COVID-19 en phase finale d'essais cliniques, comme Moderna et Pfizer avec BioNTech.

Les vaccins conventionnels utilisent des protéines ou des fragments de protéines antigènes d'un virus pour déclencher une réponse immunitaire. Pour les produire, il faut un virus vivant (qui est inactivé avant que le produit n'atteigne les patients) et un vecteur de production, comme des cellules mammaires ou des œufs de poule qui produisent de grandes quantités de protéines virales. Cette méthode de production est difficile à mettre à l'échelle, prend beaucoup de temps et l'expédition du produit final est coûteuse. D'autre part, les vaccins à ARNm utilisent les propres cellules d'une personne pour produire l'antigène de la protéine virale. L'ARNm contient des instructions pour la protéine, et les cellules autour du site du vaccin le traduiront et signaleront au système immunitaire la présence d'un envahisseur. Contrairement à un virus, l'ARNm ne peut pas se répliquer tout seul et se dégrade dans l'organisme. La production de vaccins à ARNm ne nécessite pas autant d'équipements spécialisés que les vaccins classiques, et ils peuvent être produits rapidement en cas de besoin, dans des sites du monde entier.

L'ARN est très sujet à la dégradation, mais la liaison interne des bases entre elles peut contribuer à rendre une séquence d'ARN plus stable. Comme plusieurs séquences d'ARNm peuvent produire la même protéine (en raison de l'utilisation de codons alternatifs), il y a une certaine marge de manœuvre dans la conception des ARNm.

Objectif de la compétition

Créer un modèle qui identifie les bases qui ont les liens les plus faibles dans les molécules d'ARNm.

Matériel et méthodes :

**Matériel**

**Données brutes**

Les données ont été téléchargées à partir de Kaggle et consistent en deux fichiers .json, train.json et test.json.

3029 séquences d’ARNm obtenues par les chercheurs de Stanford ont été séparés entre les données test et train.

* Train.json : 2400 séquences d’ARNm de longueur 107.
* Test.json : 629 séquences d’ARNm de longueur 107.

La compétition demandait des prévisions de base pour trois « targets » continus liés à la dégradation :

* Réactivité
* Dégradation à 50C en présence de Mg
* Dégradation à pH10 et 50C en présence de Mg

On nous demande de les prévoir en utilisant les caractéristiques fournies pour chaque ARN :

* Séquence (chaîne de bases, ou les ACTG)
* Longueur (nombre de bases)
* Structure (représentation de la structure 3D de l'ARN du serveur web RNAFold, une chaîne contenant '(', ')' et '.', où '.' représente les bases non appariées et '(' et ')' représentent les bases liées entre elles sur le côté supérieur et inférieur d'une boucle, respectivement)
* Type de structure en boucle (ceci est également tiré de RNAFold, et c'est une chaîne de caractères annotant chaque base comme faisant partie d'une tige, d'une boucle interne, d'un renflement, d'une boucle externe, etc.)
* Erreur dans la mesure des cibles

**Conda**

Conda est un puissant gestionnaire de packages et d'environnement utilisé avec les commandes de ligne de commande à l'invite Anaconda pour Windows, ou dans un terminal pour macOS ou Linux.

**Python**

Ce programme a été condé en Python3 avec un notebook JuPyteR.

**Librairies Python**

Pour exécuter ce programme nous avons utilisé des librairies non-standards pour Python :

* **P**andas
* **N**umpy
* **M**atplotlib.pyplot
* **S**eaborn
* **T**ensorflow

**Kaggle**

**Méthodes**