Basecalling par deep learning sur les données de séquençage par nanopores du MinION

Adrien Jarretier-Yuste
Encadré par Hector Hernandez-Vargas,
Chloé Goldsmith,
Céline Robardet,
Stefan Duffner,
Marc Plantevit

Université Claude Bernard Lyon 1, France

Résumé L'ADN est une macromolécule biologique présente chez tous les organismes vivants connus, l'unité de base est le nucléotide, ou "base". La séquence des bases du génome humain est essentiellement connue depuis 2003, cependant il existe encore aujourd'hui de nombreuses inconnues sur la structure de notre ADN.
Dans ce stage nous nous intéressons à la détection des modifications chimiques d'une des quatre bases de l'ADN, la cytosine, avec un séquenceur de troisième génération. Le MinION est un appareil pour le séquençage par nanopores dont la sortie est une série temporelle correspondant à la mesure d'un courant électrique. Le *basecalling* pour cet appareil est le processus de traduction de cette mesure en une séquence ADN.
L'objectif de ce stage est d'abord de mettre en place une chaîne de traitement permettant d'analyser rapidement et simplement l'ADN séquencé avec le MinION. Dans un second temps, nous proposons une méthode permettant d'effectuer un *basecalling* en différenciant directement les cytosines non modifiées et modifiées, ce faisant en considérant une cinquième base. Le signal à analyser est une série temporelle, comparable aux données issues d'un enregistrement audio. Il existe dans la littérature sur la reconnaissance automatique de la parole des modèles performants sur ces données. Le modèle choisi et implémenté est un modèle de *deep learning* de type CNN-RNN-CTC. Un réseau de neurones à convolution suivi d'un réseau de neurones récurrents de type LSTM bidirectionnel est utilisé pour effectuer une classification connexionniste temporelle. Un tel réseau peut être entraîné pour labelliser correctement une séquence ADN en fonction d'une série temporelle. Les résultats montrent que notre modèle est capable d'apprendre à aligner et restituer correctement une séquence à partir du séquençage d'un brin d'ADN synthétique de quelques centaines de bases.

Mots-clés: basecalling, bioinformatique, deep learning, séquençage par nanopores, réseau de neurones récurrents

Abstract. DNA is a biological macromolecule present in all known living organisms, the basic unit is the nucleotide, or "base". The sequence of the human genome is essentially known since 2003, however there are still today many unknowns about the structure of our DNA.
In this internship we are interested in detecting the chemical modifications of one of the four bases of DNA, cytosine, with a third-generation sequencer. The MinION is a device for nanopore sequencing whose output is a time series corresponding to a current measurement. Basecalling for this device is the process of translating this measurement into a DNA sequence.
The objective of this internship is first to set up a pipeline for quick and simple analysis of DNA sequenced with the MinION. Subsequently, we propose a method for direct discrimination of non modified and modified cytosines as a fifth base during basecalling. The signal to analyse is a time series, comparable to data from an audio recording. In speech recognition literature, there are powerful models on this kind of data. The chosen and implemented model is a deep learning model, specifically a CNN-RNN-CTC. A convolutional neural network followed by a bidirectional LSTM recurrent neural network is used to perform a connectionist temporal classification. Such a network can be trained to properly label a DNA sequence based on a time series. Results show that our model is able to learn to correctly align and return a sequence from the sequencing of a synthetic DNA strand of a few hundred bases.

Keywords: basecalling, bioinformatics, deep learning, nanopore sequencing, recurrent neural network

1 Introduction

1.1 La méthylation de l'ADN

La méthylation de l'ADN est un type de modification chimique de l'ADN, c'est un processus dans lequel un groupe méthyle est ajouté sur le 5ème carbone d'une cytosine la convertissant en 5-méthylcytosine (5mC) (Fig.1).
Il existe de nombreuses autres modifications possibles de l'ADN [33], nous nous intéressons en particulier à la 5mC et la 5hmC (5-hydroxyméthylcytosine) chez l'humain.

Chez l'humain, la méthylation de l'ADN arrive en général sur les cytosines qui précèdent une guanine (on appelle ça un site CpG). Entre 60 et 90% des sites CpG sont méthylés chez les mammifères [21], certaines régions de l'ADN ont une concentration élevée en sites CpG, on les nomme îlots CpG et sont habituellement non méthylés. Ces derniers sont le plus souvent localisés dans les séquences promotrices. Un promoteur est une région de l'ADN située en amont du site de démarrage de la transcription d'un gène [10] [30].

Lorsque un îlot CpG présent dans le promoteur est méthylé, cela est associé à l'inhibition du gène en aval (Fig.2).

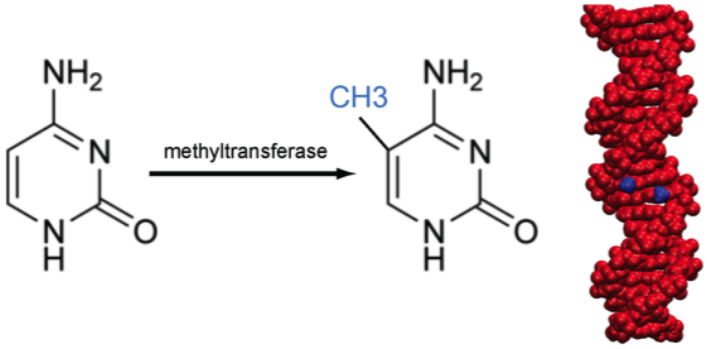

Fig. 1: Processus de méthylation (schéma de Mikhail Dozmorov)

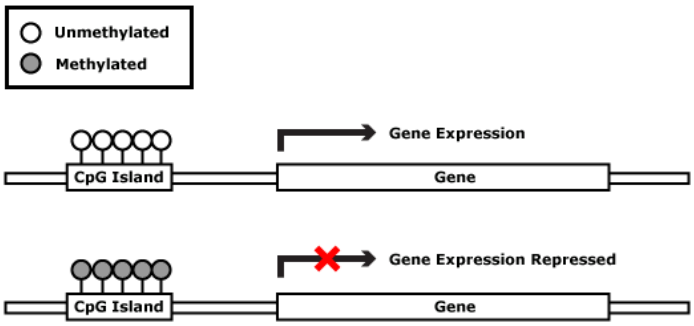

Fig. 2: La méthylation dans le promoteur est associée à l'inactivation du gène (schéma de Mikhail Dozmorov)

La méthylation est un des phénomènes déterminant l'identité des cellules, c'est ainsi que des cellules possédant la même séquence ADN de base, se différencient en types spécialisés (un exemple avec la différenciation des cellules du sang en figure 3).

Une méthylation anormale est associée à des processus cellulaires dérégulés qui peuvent entraîner un cancer [17] [18].

Les fonctions de la méthylation sont encore peu comprises. De nouvelles méthodes nous permettent d'étudier le génome en entier et de peindre un portrait plus clair du méthylome. Il semblerait que la relation entre la méthylation de l'ADN et la transcription d'un gène soit plus dynamique et nuancée que ce qui avait était anticipé [23].

La 5-hydroxymethylecytosine (5hmC) est une autre modification chimique de la cytosine, un groupe méthyle sur une cystosine (5mC) peut être oxydé en 5hmC.

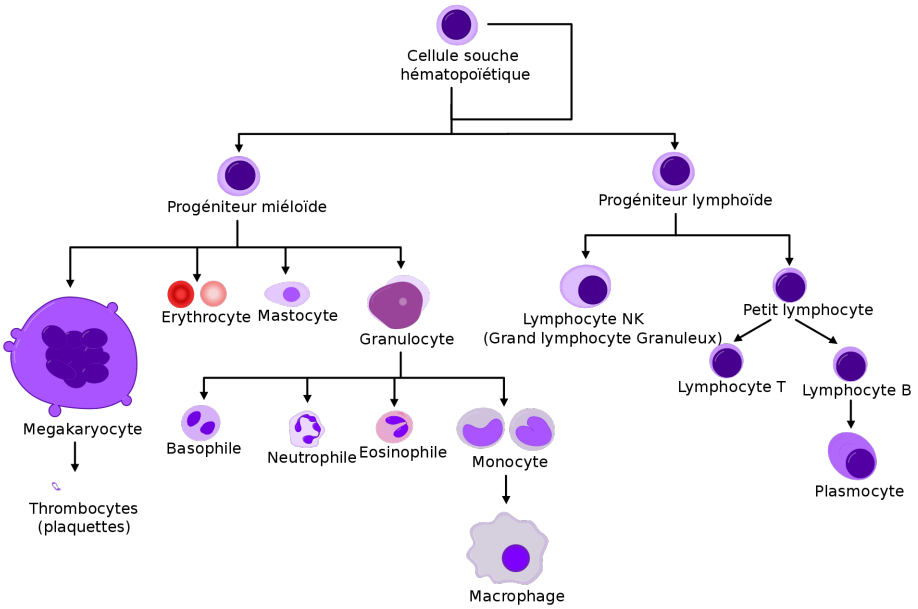

Fig. 3: Modèle simplifié de l'hématopoïèse. Une Cellule Souche Hématopoïétique multipotente se différencie en progéniteur lymphoïde ou myéloïde. La différenciation du progéniteur lymphoïde aboutit aux différents types de lymphocytes. La différenciation du progéniteur myéloïde fournit après plusieurs étapes les érythrocytes, les autres leucocytes (granulocytes, monocytes et macrophages) ainsi que les plaquettes. (Traduit par DrGMur de Mikael Häggström, de l'original par A. Rad)

5hmC a récemmement été décrite comme la "6ème base" de l'ADN, les premières recherches indiquent que son rôle fonctionnel est distinct de la 5mC avec un rôle déterminant dans la différentiation des cellules progénitrices [11].

1.2 La plasticité du système immunitaire

Notre système immunitaire, est une système biologique complexe caractérisé par sa flexibilité. Les cellules cancéreuses peuvent être détectées et détruites ou bien laissées à se diviser et proliférer. Cette réponse immunitaire est coordonnée par différents types cellulaires parmi lesquels font partie les cellules T qui sont connues pour jouer un rôle clé. L'équilibre entre les sous-types de cellules T (par exemple, Th0, Tregs, Th17) influence le résultat final de la réponse [3] [36]. Ces sous-types de cellules T ne sont pas fixés, leur type cellulaire est connu pour être "plastique" [38], et leur changement de proportions est dans certains cas associé au cancer.

Fig. 4: Le *squiggle* est le tracé du signal mesuré par le MinION. Pour l'illustration, une séquence après *basecalling* a été ajoutée au dessus, pour une visualisation d'un véritable signal voir figure 11

Ce stage s'inscrit dans le projet AEROBICS (*Analysis and Epigenetic Recognition Of dysBalanced Immune Cell plaSticity*) qui vise à étudier et mesurer ce déséquilibre de cellules immunitaires comme marqueur de cancer en identifiant les modifications de méthylation. Ce projet fait lui-même partie du plus grand projet PLASCAN (Prévenir la plasticité et l'adaptabilité tumorale : vers la nouvelle génération de médecine personnalisée), un projet de recherche multidisciplinaire pour la compréhension et la modélisation du Cancer.

1.3 Le séquençage ADN par nanopores

Afin de mieux comprendre ces mécanismes, nous cherchons à étudier l'ADN et ses modifications à l'aide d'un séquenceur de troisième génération, le MinION, développé et distribué par Oxford Nanopore Technologies depuis 2014 [22].

Oxford Nanopore Technologies (ONT) est une société basée au Royaume-Uni qui développe des produits de séquençage par nanopores (y compris le séquenceur d'ADN portable MinION).

Le concept de base du séquençage par nanopores est le suivant :
Faire passer un brin d'ADN à travers un pore à l'échelle nanométrique d'une membrane de la tête aux pieds et appliquer une différence de potentiel à travers la membrane. Les nucléotides présents dans le pore affecteront sa résistance électrique de sorte que les mesures de courant dans le temps puissent indiquer la séquence de bases de l'ADN traversant le pore [24] [5]. Ce signal de courant électrique (alias *squiggle* ou "gribouillis" en français dû à son apparence lors du tracé) correspond aux données brutes recueillies par un séquenceur ONT (Fig. 4).

1.4 Le basecalling

Le *basecalling* pour les dispositifs ONT est le processus de traduction de ce signal brut en une séquence ADN. Ce processus n'est pas une tâche triviale car les signaux électriques proviennent de molécules uniques et la vitesse à laquelle le brin d'ADN traverse le pore est non uniforme, ce qui produit des données bruitées [29]. En outre, la résistance électrique d'un pore est déterminée par les bases présentes dans plusieurs nucléotides qui se trouvent dans le point le plus étroit du pore (environ cinq nucléotides pour les pores R9.4), ce qui donne un grand nombre d'états possibles (ou k-mers) :
$4^5 = 1024$ pour un modèle standard à quatre bases.
Lorsque des bases modifiées sont présentes, par ex. 5-méthylcytosine, le nombre d'états possibles peut encore augmenter :
$5^5 = 3125$.
Du fait de la non-uniformité de la vitesse de transition de l'ADN à travers le nanopore, il est également difficile de détecter une transition entre deux k-mers identiques [29]. Cela fait du *basecalling* des appareils ONT un problème de *machine learning* difficile et un facteur déterminant pour la qualité et l'exploitabilité du séquençage ONT.

Durant ce stage nous avons deux objectifs :

1. Mettre en place un *pipeline* permettant d'analyser rapidement et simplement l'ADN séquencé avec le MinION (Fig. 6).
2. Développer un modèle de *deep learning* pour effectuer le *basecalling* en différenciant directement les cytosines non modifiées et modifiées. Ce modèle pourra être entraîné spécialement pour l'ADN humain avec la prise en compte des cytosines méthylées.

2 Le pipeline d'analyse

Afin d'analyser efficacement les données fournies par le MinION (Fig. 6) nous développons un *pipeline* avec différents outils existants.

Dans la suite de ce document nous parlerons d'un "ADN standard", c'est un ensemble de trois échantillons d'ADN synthétique court (900 bases) fournit par Zymo Research (https://www.zymoresearch.com/products/5-methylcytosine-5-hydroxymethylcytosine-dna-standard-set) pour lequel nous connaissons parfaitement la séquence, nous avons séquencé ces échantillons avec le MinION afin d'avoir des données de contrôle pour l'ensemble des expériences.

Ces trois échantillons ont exactement la même séquence de bases mais les cytosines ont été modifiées différemment :

- Le premier échantillon ne contient que des cytosines sans modification.
- Le deuxième échantillon contient uniquement des 5-méthylcytosines.
- Le troisième échantillon contient 100% de 5-hydroxyméthylcytosines.

2.1 Description du pipeline

Le *pipeline* final qui a été développé est représenté en figure 7.

- Le *basecaller* effectue le *basecalling* (cf 1.4), le résultat de cette premiere partie d'analyse est un ensemble de fichiers au format fastQ [8], ces fichiers contiennent, pour chaque brin d'ADN qui est passé dans un nanopore, sa séquence accompagnée de scores de qualité phred [13] [12].

- Avant le séquençage chaque échantillon d'ADN est "barcodé", un code barre [25] artificiel est attaché à chaque brin d'ADN. Ainsi on peut séquencer plusieurs échantillons en même temps. Une fois la séquence identifiée après le *basecalling* on peut séparer les différents échantillons avec le démultiplexeur. Cette étape nous permet de séparer les fichiers fastQ dans différents dossiers, chacun correspondant à un code barre spécifique.

- Puis il faut aligner les séquences avec un fichier de référence, pour l'ADN standard la séquence est fournie par Zymo Research, pour l'humain on utilise hg38. Cet phase d'alignement nous donne des fichiers SAM (*Sequence Alignment Map*) qui contiennent les séquences obtenues précédemment avec leur position dans le génome.

- Enfin, on peut identifier la fréquence de méthylation de régions du génome à l'aide de Nanopolish [32].

2.2 Le choix du basecaller

Il existe plusieurs outils capables d'effectuer le *basecalling* :

- Albacore (disponible sur `community.nanoporetech.com`), l'ancien *basecaller* développé par ONT, maintenant remplacé par Guppy
- BasecRAWller [34],
- Chiron [35], un *basecaller* à base de *deep learning* similaire au modèle que nous implémentons par la suite, un CNN-RNN-CTC.
- DeepNano [4]
- Flappie (`https://github.com/nanoporetech/flappie`) de ONT, un *basecaller* expérimental qui utilises un CTC, il a remplacé Scrappie.
- Guppy (disponible sur `community.nanoporetech.com`), le *basecaller* actuel de ONT à base de réseau Gated Recurrent Unit.
- Nanocall [9]
- Scrappie (`https://github.com/nanoporetech/scrappie`) de ONT, anciennement un *basecaller* expérimental, maintenant remplacé par Flappie

Leurs performances ont été comparées dans la littérature [37] [29]. Ces résultats montrent empiriquement que les nouveaux *basecallers* à base de réseaux de neurones surpassent les anciens modèles HMM (Hidden Markov Models). Ainsi Oxford Nanopore Technologies intègrent rapidement dans leur *basecaller*

les modèles les plus efficaces et les plus à même de convenir aux évolutions de version des nanopores du MinION.

Leur *basecaller* Guppy, qui utilise un réseau récurrent GRU, a donc été notre choix pour notre *pipeline*.

Guppy est fournit avec plusieurs parties, *Guppy_ basecaller*, *Guppy_ barcoder* et *Guppy_ aligner*.

Pour la détection de la méthylation nous avions le choix entre SignalAlign [28] et Nanopolish [32]. Les 2 programme sont des HMM, cependant, SignalAlign fonctionne sous python 2 et n'a pas été mis à jour depuis 3 ans, nous avons donc préféré utilisé Nanopolish.

2.3 Résultat du pipeline

L'utilisation de notre *pipeline* sur les échantillons d'ADN standard montre qu'il nous est possible de différencier les séquences globalement méthylées des séquences non modifiées (Tab. 1, Fig. 5).

De plus l'analyse de l'échantillon 5-hydroxyméthylé (5hmC) nous permet de vérifier que cette modification n'est pas détectée en faux positif pour la 5mC (Tab. 1, Fig. 5). Nanopolish [32] n'est pas entraîné pour la 5hmC, il est normal qu'il ne détecte pas la modification chimique sur cet échantillon.

Echantillon	Non modifié	5-méthylé	5-hydroxyméthylé
Nombre de sites CpG	2391329	10521	3485
Nombre de sites CpG 5-méthylés	33348	6939	64
P-value	$< .000001$	$< .000001$	$< .000001$
Méthylation moyenne	.014	.660	.019

Table 1: Taux de méthylation moyens des sites CpG des différents échantillons d'ADN standard obtenus avec notre *pipeline*.

Malgré les résultats positifs de l'exécution de notre *pipeline*, deux points peuvent être immédiatement soulevés en voyant ces résultats :

1. Pour la méthylation moyenne, nous attendons 0, 1 et 0 or nous obtenons 0.014, 0.660 et 0.019, jusqu'à 34% d'erreur dans la détection des Cytosines réellement méthylées.
2. Sur le nombre de sites CpG, entre les trois échantillons d'ADN synthétiques, les séquences sont exactement les mêmes, nous sommes donc en mesure d'attendre environ 1/3 de sites CpG dans chaque échantillon, en revanche nous obtenons une distribution du nombre de sites entre les échantillons C, 5mC et 5hmC de 99%, 0.4% et 0.1%.

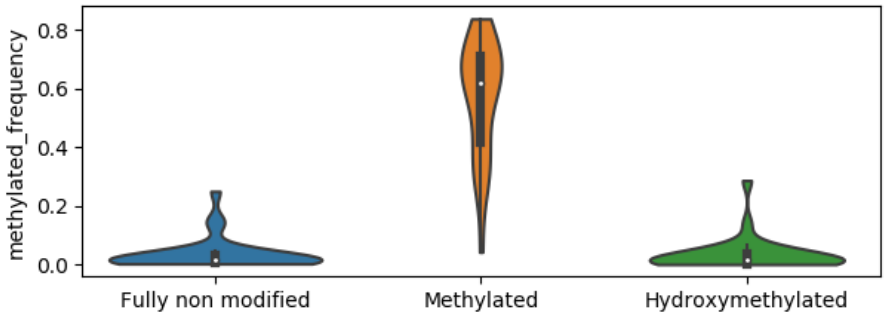

Fig. 5: Fréquences de cytosines 5-méthylées détectées par Nanopolish à la fin du *pipeline* sur les échantillons d'ADN standard. Les premier et troisième diagrammes correspondent respectivement à l'échantillon d'ADN complètement non modifié et à l'échantillon d'ADN 5-hydroxyméthylé, ils sont correctement détectés comme très peu 5-méthylés. Le deuxième échantillon correspond à l'ADN standard contenant des cytosines 5-méthylées, Nanopolish le détecte comme bien plus fortement méthylé que les autres.

Dans notre *pipeline*, les sources d'erreurs durant l'analyse sont nombreuses et cumulables. Durant le *basecalling* des erreurs de prédiction des séquences peuvent survenir, une telle erreur peut être répercutée dans la phase de démultiplexage si le code barre a été labellisé incorrectement. Enfin Nanopolish peut également faire des erreurs en classifiant un site CpG comme méthylé ou non.

De plus, le modèle utilisé par Guppy est paramétré et entraîné pour la génération de séquences avec 4 labels (A, C, G et T), il n'est pas entraîné pour reconnaître la 5mC ou la 5hmC. Il a été démontré qu'il est possible de détecter la 5-hydroxyméthylcytosine distinctement de la 5-méthylcytosine [26], le courant électrique mesuré est différent lorsque les cytosines sont non modifiées, 5-méthylées ou encore 5-hydroxyméthylées. Ceci explique l'écart du nombre de sites entre les différents échantillons, les séquences des échantillons modifiés sont très probablement mal classifiées et considérées comme de mauvaise qualité à l'étape du *basecalling*.

Dans le but d'étudier la méthylation et l'hydroxyméthylation nous proposons par la suite de développer notre modèle afin de pouvoir l'entraîner pour notre besoin de *basecalling*. En ayant la maîtrise du modèle nous pourrons le paramétrer et l'entraîner pour labelliser les sites méthylés et hydroxyméthylés.

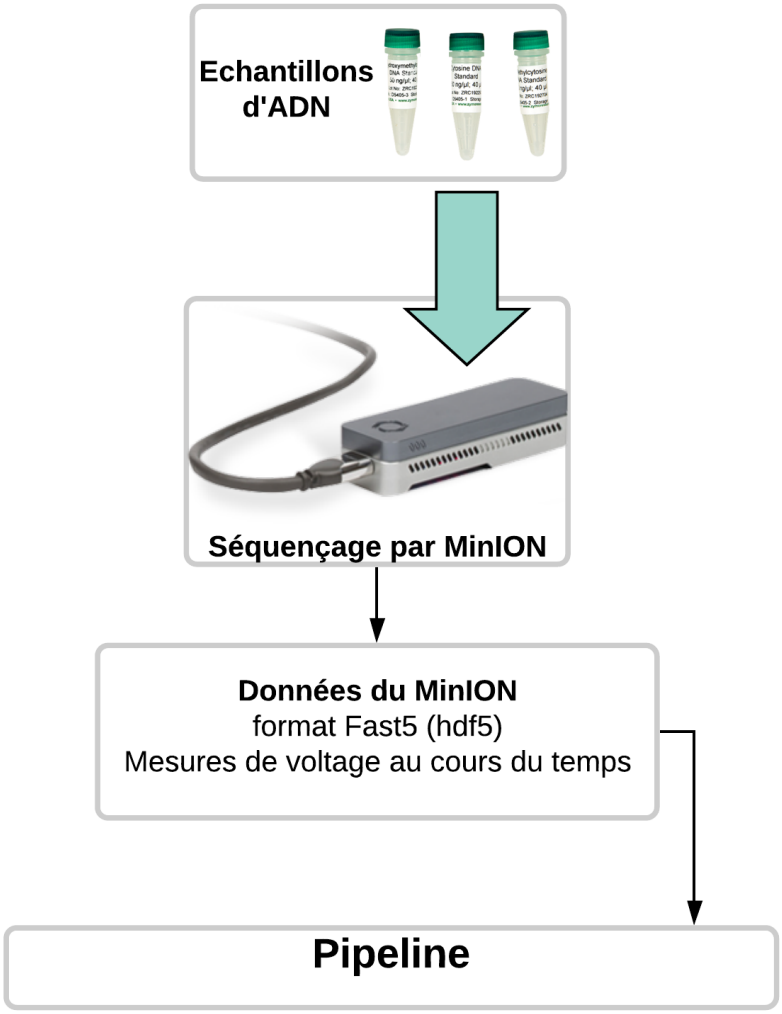

Fig. 6: Le séquençage de l'ADN avec le MinION, les échantillons d'ADN à analyser sont préparés, un code barre est ajouté à chacun (multiplexage). Une fois la préparation terminée, les échantillons sont insérés dans le séquenceur. Les lectures de séquençage seront analysées avec notre *pipeline* (Fig. 7).

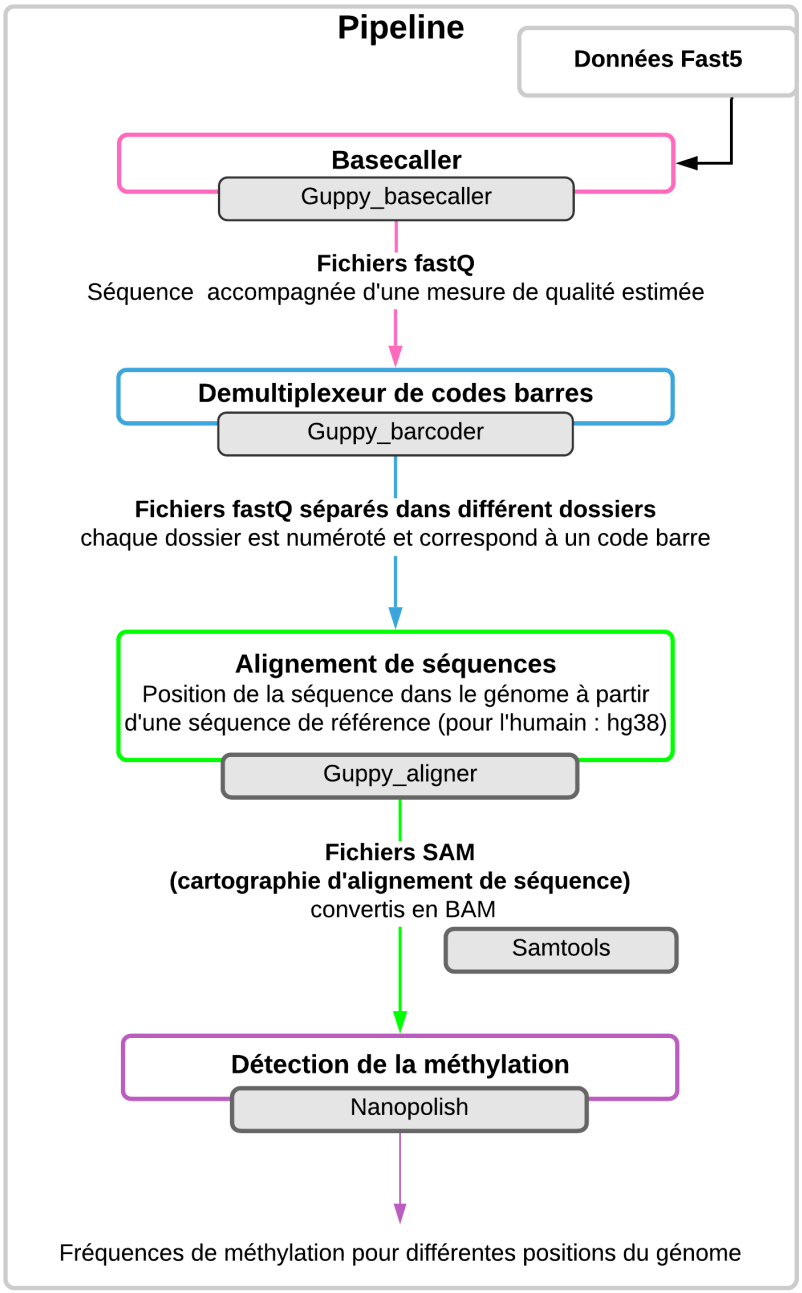

Fig. 7: Le *pipeline* d'analyse des données du MinION (cf Fig. 6).

3 L'implémentation d'un basecaller sur mesure, modèle CNN-RNN-CTC

En plus des problèmes vus précédemment inhérents au *basecalling* du signal du MinION (1.4), nous avons les connaissances suivantes sur l'échantillonnage du signal électrique :

- La fréquence d'échantillonnage est de 4 kHz.
- Un brin d'ADN traverse un nanopore en moyenne à 450 bases / s, cette vitesse varie en fonction de la composition en nucléotides, la protéine motrice peut également arrêter de fonctionner si elle n'a plus d'énergie (ATP).

Nous avons donc en moyenne 9 mesures de courant par k-mer.

Cet échantillonage est d'une forme similaire à ce que l'on retrouve dans la reconnaissance automatique de la parole [16].

3.1 La prédiction de séquences avec le modèle CTC

Le modèle CTC (pour *Connectionist Temporal Classification*) est une modification apportée à un réseau de neurones récurrents pendant l'apprentissage pour lui permettre d'apprendre à classifier une séquence à partir d'une série de données temporelles [15] sans étape de segmentation au préalable.

L'idée de base du CTC est la suivante : pour pouvoir prédire une séquence z avec l'alphabet finit L à partir d'une série de données x avec $|z| \leq |x|$, on rajoute à L un label *blank*, $L' = L \cup blank$.

1. Pour chaque entrée X_t, le réseau de neurones récurrents donne une prédiction.
2. Chaque prédiction est transformée en vecteur de probabilités de taille $|L'|$.
3. La matrice obtenue est ensuite décodée pour obtenir une séquence de labels de L'.
4. Les répétitions consécutives de labels sont remplacées par ce label une seule fois, cela correspond dans notre cas aux multiples mesures d'un même k-mer alors que le brin d'ADN ne s'est pas déplacé.
5. Les *blanks* sont supprimés, ils indiquent les changements de k-mer, cet étape combinée à la précédente permet de différencier les nucléotides consécutifs identiques.

   Exemple :
   (a) Pour une entrée de 20 mesures,
   la séquence décodée est _AAGG_CC_GGTTTGG__GG
   (b) Répétitions consécutives supprimées : _AG_C_GTG_G
   (c) *Blanks* supprimés : AGCGTGG
6. Comparer la séquence obtenue avec celle attendue.

Algorithm 1 CTC Beam Search

Require: Matrice de probabilités mat, $largeurBeam$
$beams \leftarrow \{\varnothing\}$
$scores(\varnothing, 0) \leftarrow 1$
for $t = 1...T$ **do**
  $meilleursBeams \leftarrow meilleursBeams(beams, largeurBeam)$
  $beams \leftarrow \{\}$
  for $b \in meilleursBeams$ **do**
    $beams \leftarrow beams \cup b$
    $scores(b, t) \leftarrow calculerScore(mat, b, t)$
    for $c \in alphabet$ **do**
      $b' \leftarrow b + c$
      $scores(b', t) \leftarrow calculerScore(mat, b', t)$
      $beams \leftarrow beams \cup b'$
    end for
  end for
end for
return $meilleursBeams(beams, 1)$

3.2 Le décodage de la matrice, l'algorithme Beam search

L'étape 3 dans la prédiction avec le modèle CTC, le décodage, peut se faire de deux façons. La première méthode, la plus simple et la plus rapide, est gloutonne, elle consiste simplement à sélectionner les labels les plus probables à chaque pas de temps (Fig. 8).

Cette méthode a cependant un inconvénient, comme elle ne prend pas en compte l'ensemble des possibilités de chemins dans la matrice, on peut obtenir une séquence dont la probabilité est sous-optimale, c'est pourquoi il peut être utile de visiter différents chemins (Fig. 9). Cependant il est impossible de visiter exhaustivement tous les chemins, la complexité est trop importante, un algorithme a été proposé, le *Beam Search* [16] qui permet d'itérer sur les pas de temps en gardant une mémoire de taille paramétrable sur les meilleurs chemins visités (Alg. 1).

3.3 Entraînement du modèle

L'entraînement d'un CTC vise à maximiser le logarithme des probabilités de classifications correctes sur le jeu de données d'entraînement, formellement défini comme suit :

Avec S l'ensemble des données d'entraînement, $(x, z) \in S$,
x les données d'entrée, z les séquences attendues, la fonction d'objectif à minimiser est :

$$-\sum_{(x,z)\in S} ln(p(z|x)) \tag{1}$$

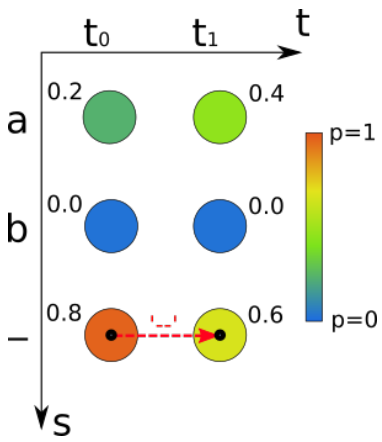

Fig. 8: Décodage de la matrice de probabilités par la méthode gloutonne, à chaque pas de temps le label de plus grande probabilité est choisi. La séquence est "", La probabilité du chemin est $0.8 * 0.6 = 0.48$. (Illustration provenant de `https://towardsdatascience.com/`)

Nous avons implémenté le modèle avec l'API Keras [7] et le backend TensorFlow [1]. Le modèle, représenté en Figure10, est composé d'une première couche de réseau convolutionnel (CNN), suivie d'un réseau de neurones récurrents (RNN).

Cette architecture CNN-RNN est classique dans le domaine de la reconnaissance automatique de la parole [2] lors de la prédiction de séquences de bout en bout sans segmentation. De plus il a été démontré sur des données de MinION que cette architecture donne de meilleures prédictions qu'un réseau CNN ou RNN seul [35].

Le RNN est un réseau bidirectionnel [31] à LSTM [19] [14].

Le RNN fournit une prédiction à chaque pas de temps, la séquence de prédictions résultante est passée à un perceptron dont la fonction d'activation est un softmax [6].

Le résultat de ce softmax est une matrice de prédictions dont le décodage est décrit en 3.2.

Préparation des données d'entraînement L'entraînement des modèles de *basecalling* présente une difficulté majeure, il est impossible de fournir au modèle un ensemble de données d'entraînement S dans lequel nous connaissons parfaitement la correspondance entre x et z. En effet même en connaissant la séquence des échantillons d'ADN standard nous ne sommes par exemple pas garanti qu'une lecture obtenue par le MinION ne soit pas une lecture partielle, ce qui peut arriver si le brin d'ADN se brise.

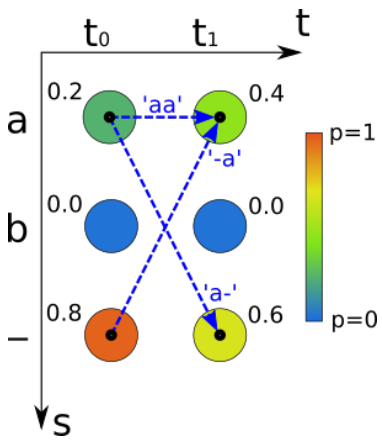

Fig. 9: Décodage en testant tous les chemins, les probabilités des chemins qui donnent la même séquence sont additionnées. La séquence ayant la plus grande probabilité est sélectionnée. Les 3 chemins qui donnent "a" ont pour probabilités : $0.2 * 0.4 = 0.08$, $0.2 * 0.6 = 0.12$, $0.8 * 0.4 = 0.32$, leur somme est égale à 0.52, $0.52 > 0.48$ donc la séquence "a" est plus probable que la séquence "". (Illustration provenant de `https://towardsdatascience.com/`)

La méthode pour créer notre jeu de données est dès lors quelque peu circulaire, elle consiste à séquencer les échantillons d'ADN avec un *basecaller* que l'on considère fiable, puis d'utiliser ces données comme base d'apprentissage.

Oxford Nanopore Technologies fournit un outil qui nous permet de traiter les données de séquençage, brutes (fastA), et *basecallées* (fastQ). Taiyaki (`https://github.com/nanoporetech/taiyaki`) nous permet d'effectuer la correspondance entre un signal de courant mesuré et la séquence que nous obtenons avec notre *pipeline*. Nous pouvons ainsi obtenir un fichier dans lequel un ensemble de données x est "mappé" aux séquences z correspondantes. Une visualisation de parties de deux séquences alignées ainsi sur leur signal est visible en figure 11.

3.4 Résultats

Nous avons réalisé une série d'entraînements afin d'évaluer les capacités d'apprentissage du modèle et mettre à l'épreuve différentes hypothèses.

Pour notre premier entraînement nous utilisons 50 séquences qui seront réparties en 40 / 10 pour les jeux d'entraînement / validation (Fig. 12). Nous observons que l'entraînement converge très rapidement. L'erreur ne descend pas en dessous de 800 qui est une très grosse valeur.

– Hypothèse : Le nombre de séquences n'a pas d'influence sur la valeur de l'erreur.

- Expérience : Entraîner le modèle avec un très petit nombre de séquences pour mettre en évidence une différence de valeur de l'erreur (Fig. 13).
- Résultat : Avec seulement 4 séquences nous obtenons déjà une erreur proche de 800, donc pas de changement par rapport au jeu de données avec 40 séquences.

- Hypothèse : L'erreur ne provient pas d'un biais dans le jeu de données.
- Expérience : Répéter l'entraînement avec des ensemble de séquences disjoins (Fig. 14).
- Résultat : Nous obtenons systématiquement des erreurs similaires, ce qui tendrait à faire penser que le problème ne vient pas des données en elles-même.

- Hypothèse : La longueur des séquences n'a pas d'influence sur la valeur de l'erreur.
- Expérience : Nous entraînons successivement le modèle avec 3 jeux de 4 séquences dont la longueur est contrainte (Fig. 15).
- Résultat : On observe une différence forte de l'erreur corrélée à la longueur moyenne des séquences, l'hypothèse peut être rejetée, le calcul de l'erreur est biaisé par la longueur de la séquence.

Nous avons dernièrement entraîné notre modèle sur un jeu de
données d'entraînement ne contenant qu'une seule séquence pendant plus de 8 heures, soit 6600 itérations. Comme nous l'attendions, l'erreur sur la séquence d'entraînement diminue mais n'ayant qu'un seul exemple pour apprendre, le modèle *overfit* et l'erreur sur la validation ne fait qu'augmenter (Fig. 16). La valeur finale de l'erreur est 0.2748, une distance de Levenshtein entre la séquence prédite et la séquence attendue nous donne 973, mais si nous supprimons les 973 dernier labels de notre prédiction, alors nous avons une distance de 0. La séquence prédite est exactement celle attendue, le modèle a bien appris par coeur, cependant des labels supplémentaires sont rajoutés à la fin et faussent la prédiction.

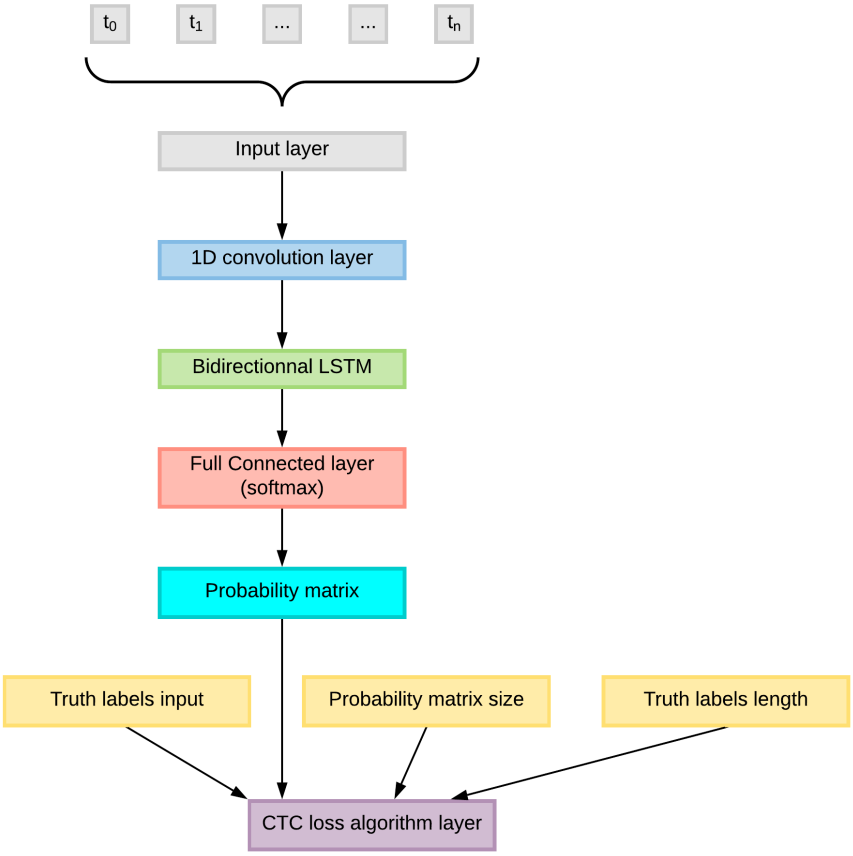

Fig. 10: Le modèle implémenté. Une série temporelle est donnée en entrée, la première couche, un réseau convolutionnel, extrait les motifs locaux. Le réseau récurrent émet une prédiction pour chaque pas de temps. La couche suivante est un perceptron dont la fonction d'activation est un softmax [6], la sortie est alors une matrice de dimensions (n, $|L'|$) contenant pour chaque pas de temps la probabilité prédite de chaque label de L'. Enfin la dernière couche reçoit la séquence attendue, la matrice de probabilités, le nombre de données (n), la taille de la séquence attendue, et calcule l'erreur CTC.

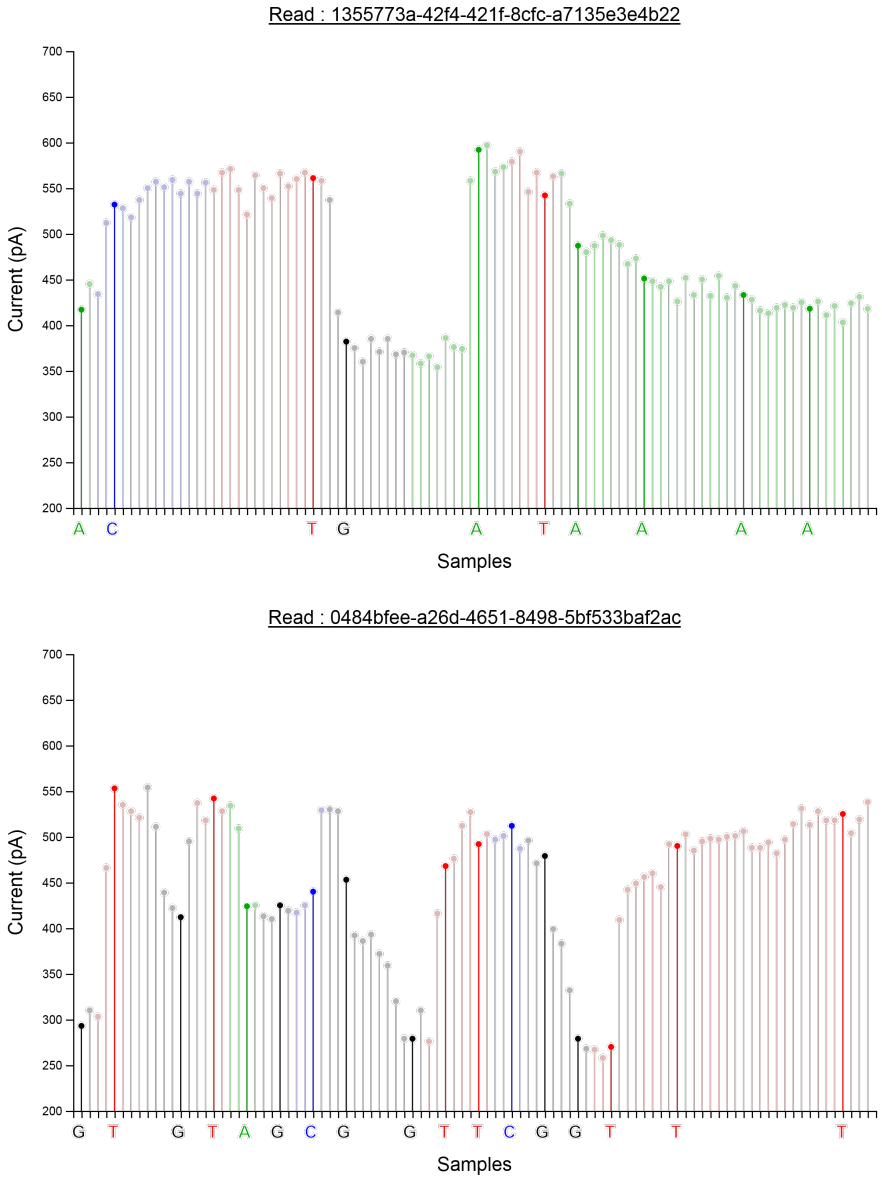

Fig. 11: Séquences d'ADN Standard non méthylées labellisées, alignées sur leur *squiggle*, résultats obtenus avec Taiyaki après avoir utilisé Guppy pour le *base-calling*.

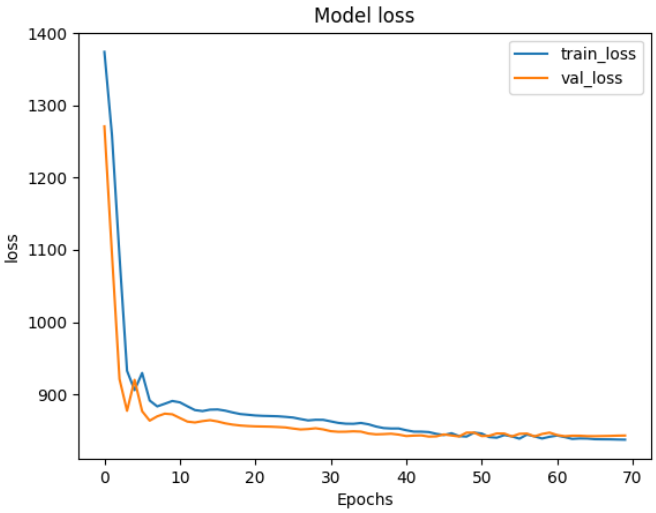

Fig. 12: Evolution de l'erreur avec 40 séquences d'entraînement et 10 de validation.

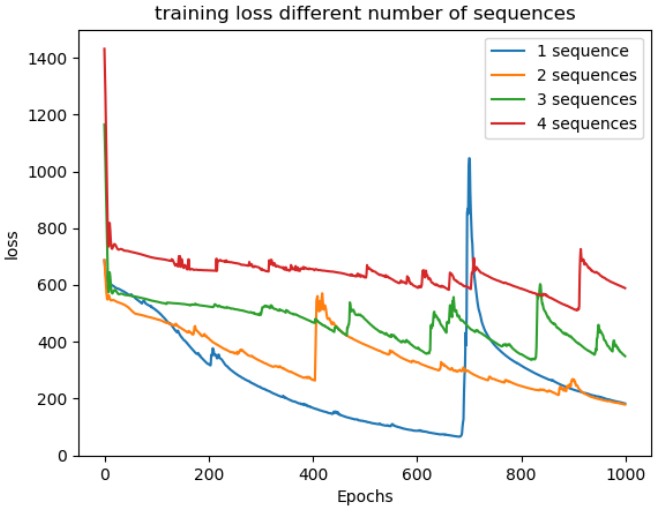

Fig. 13: Evolution de l'erreur avec des nombres de séquences différents dans les jeux de données d'entraînement.

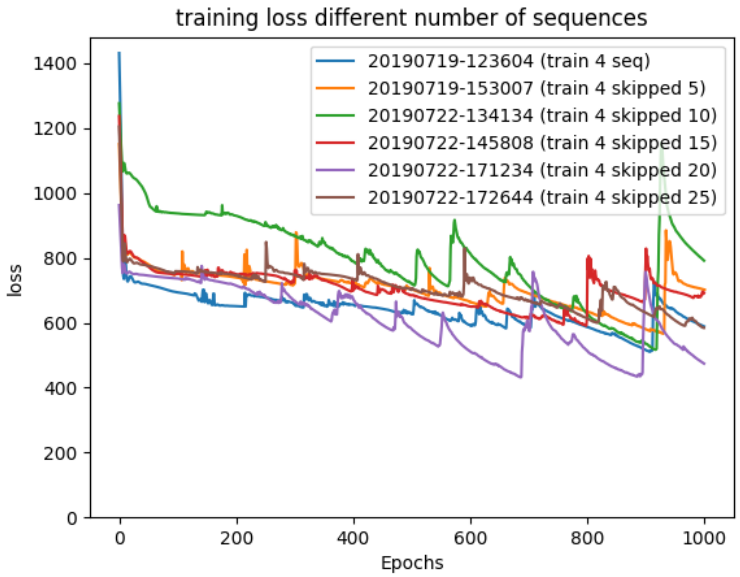

Fig. 14: Evolution de l'erreur avec des jeux d'entraînement disjoints.

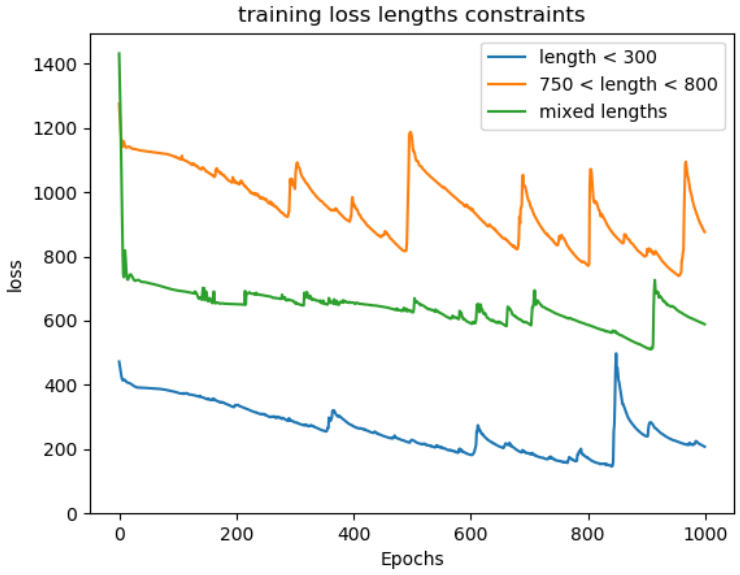

Fig. 15: Evolution de l'erreur avec des séquences de longueurs différentes.

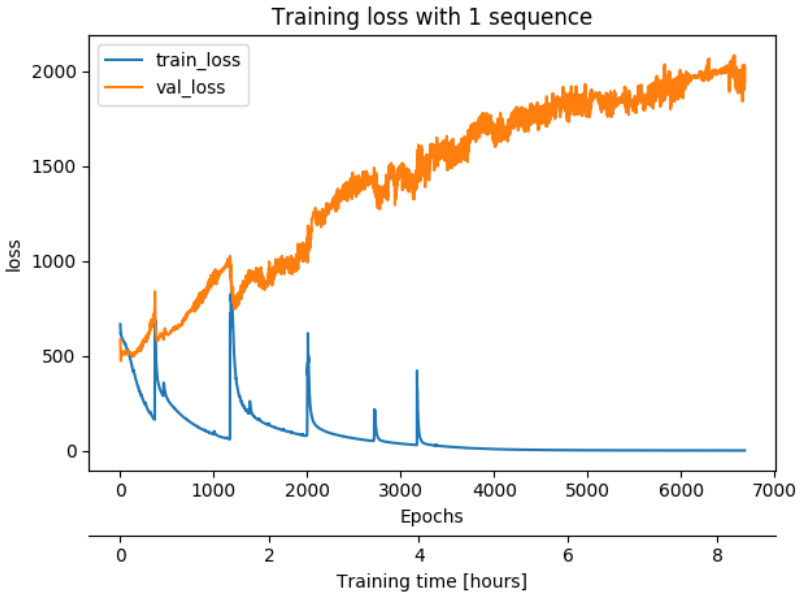

Fig. 16: Evolution de l'erreur au cours de l'entraînement sur une seule séquence, la validation est effectuée sur une séquence différente. On observe un *overfitting* attendu.

4 Discussion

Notre modèle est capable d'apprendre une séquence par coeur, cependant nous avons vu que lors de la prédiction il rajoute des labels après la séquence, ce qui fausse le résultat, l'apprentissage est également extrêmement long, 6600 itérations et 8 heures pour apprendre par coeur une séquence.

Ces problèmes sont probablement liés à des bugs dans l'implémentation et possiblement des biais dans l'apprentissage. L'implémentation a été faite à haut niveau afin de faire un protoype en peu de temps en utilisant la fonction d'optimisation présente dans TensorFlow [1]. Nous recommandons de modifier la boucle d'entraînement afin d'avoir un contrôle absolu, en particulier sur la fonction d'optimisation.

Pour accélérer l'apprentissage il faudra implémenter la normalisation par batch [20], dans un réseau de *deep learning* une normalisation entre chaque couche permet d'accélérer considérablement la convergence de l'apprentissage.

Dans le cadre de la recherche pour la compréhension des mécanismes du cancer chez l'Homme nous nous intéressons tout particulièrement à l'étude de l'ADN humain, la maîtrise du modèle nous permettra de l'entraîner sur des séquences de génome humain, ce qui en fera un *basecaller* spécialisé et non généraliste comme les *basecallers* existants. En effet le contexte génomique est complètement différent entre toutes les espèces vivantes, les bactéries, par exemple, ont des modifications chimiques très différentes de ce que l'on trouve chez l'humain.

Pour la détection de la 5hmC, il est documenté que les tissus de cerveau sont riches en régions hydroxyméthylées [27]. Pour générer une base de données d'apprentissage afin d'entraîner le modèle à reconnaître cette modification il faudra passer par des techniques comme *cas9-targeting* qui permet de cibler, pour le séquençage par nanopores, des régions connus de l'ADN.

Remerciements Je tiens à remercier tout particulièrement
Hector Hernandez-Vargas et Chloé Goldsmith pour leur encadrement au sein du laboratoire, leur disponibilité et tout le temps qu'ils ont passé
à m'enseigner énormément de choses sur la biologie, le séquençage ADN et tous les autres sujets dont nous avons pu discuter. Grâce à eux j'ai pu m'intégrer dans un projet multidisciplinaire et acquérir des connaissances incroyables sur un champ de recherche qui m'était inconnu.

Un grand merci à Céline Robardet, Stefan Duffner et Marc Plantevit de l'équipe "dm2l" du LIRIS pour leur aide et leurs conseils.

J'adresse également mes remerciements à Anthony Ferrari qui a pris du temps pour me fournir l'accès au cluster de calcul de la plateforme bioinformatique.

Je suis très reconnaissant à toute l'équipe "TGF-beta et régulation de la réponse immunitaire" du CRCL, pour leur accueil, Julien Marie, Saidi Soudja, Vincent Flacher, Alexandra Lainé, Ossama Labiad,
Ramdane Igalouzene, Olivier Fesneau, Apostol Apostolov, Sarah Bettini.

Merci également à Yenkel Grinberg-Bleyer, Robert Dante et Mounia Defontaine.

Ma sympathie va enfin à Shiqiang Xu , Inès Nihal El Rifai, Julien Laurencin, Clara Poncet, ainsi que les autres stagiaires avec qui j'ai pu échanger durant ce stage.

La rencontre de ces personnes, chercheurs, post-docs, thésards, stagiaires, m'a enseigné beaucoup de choses, au delà de la mission du stage, sur le monde de la recherche, mais aussi sur différentes cultures.

Enfin un merci à Davide Bolognini, developpeur de NanoR avec qui j'ai pu échanger et qui a été extrêmement réactif.

References

1. Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G.S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., Levenberg, J., Mané, D., Monga, R., Moore, S., Murray, D., Olah, C., Schuster, M., Shlens, J., Steiner, B., Sutskever, I., Talwar, K., Tucker, P., Vanhoucke, V., Vasudevan, V., Viégas, F., Vinyals, O., Warden, P., Wattenberg, M., Wicke, M., Yu, Y., Zheng, X.: TensorFlow: Large-scale machine learning on heterogeneous systems (2015), `https://www.tensorflow.org/`, software available from tensorflow.org
2. Amodei, D., Ananthanarayanan, S., Anubhai, R., Bai, J., Battenberg, E., Case, C., Casper, J., Catanzaro, B., Cheng, Q., Chen, G., Chen, J., Chen, J., Chen, Z., Chrzanowski, M., Coates, A., Diamos, G., Ding, K., Du, N., Elsen, E., Engel, J., Fang, W., Fan, L., Fougner, C., Gao, L., Gong, C., Hannun, A., Han, T., Johannes, L., Jiang, B., Ju, C., Jun, B., LeGresley, P., Lin, L., Liu, J., Liu, Y., Li, W., Li, X., Ma, D., Narang, S., Ng, A., Ozair, S., Peng, Y., Prenger, R., Qian, S., Quan, Z., Raiman, J., Rao, V., Satheesh, S., Seetapun, D., Sengupta, S., Srinet, K., Sriram, A., Tang, H., Tang, L., Wang, C., Wang, J., Wang, K., Wang, Y., Wang, Z., Wang, Z., Wu, S., Wei, L., Xiao, B., Xie, W., Xie, Y., Yogatama, D., Yuan, B., Zhan, J., Zhu, Z.: Deep Speech 2 : End-to-End Speech Recognition in English and Mandarin. In: International Conference on Machine Learning. pp. 173–182 (Jun 2016), `http://proceedings.mlr.press/v48/amodei16.html`
3. Basu, R., Hatton, R.D., Weaver, C.T.: The Th17 family: flexibility follows function. Immunological reviews 252(1), 89–103 (Mar 2013), `https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3607325/`
4. Boža, V., Brejová, B., Vinař, T.: DeepNano: Deep recurrent neural networks for base calling in MinION nanopore reads. PLOS ONE 12(6), e0178751 (Jun 2017), `https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0178751`
5. Branton, D., Deamer, D.W., Marziali, A., Bayley, H., Benner, S.A., Butler, T., Di Ventra, M., Garaj, S., Hibbs, A., Huang, X., Jovanovich, S.B., Krstic, P.S., Lindsay, S., Ling, X.S., Mastrangelo, C.H., Meller, A., Oliver, J.S., Pershin, Y.V., Ramsey, J.M., Riehn, R., Soni, G.V., Tabard-Cossa, V., Wanunu, M., Wiggin, M., Schloss, J.A.: The potential and challenges of nanopore sequencing. Nature biotechnology 26(10), 1146–1153 (Oct 2008), `https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2683588/`

6. Bridle, J.S.: Probabilistic Interpretation of Feedforward Classification Network Outputs, with Relationships to Statistical Pattern Recognition. In: Soulié, F.F., Hérault, J. (eds.) Neurocomputing. pp. 227–236. NATO ASI Series, Springer Berlin Heidelberg (1990)
7. Chollet, F., et al.: Keras. https://keras.io (2015)
8. Cock, P.J.A., Fields, C.J., Goto, N., Heuer, M.L., Rice, P.M.: The Sanger FASTQ file format for sequences with quality scores, and the Solexa/Illumina FASTQ variants. Nucleic Acids Research 38(6), 1767–1771 (Apr 2010), https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2847217/
9. David, M., Dursi, L.J., Yao, D., Boutros, P.C., Simpson, J.T.: Nanocall: an open source basecaller for Oxford Nanopore sequencing data. Bioinformatics 33(1), 49–55 (Jan 2017), https://academic.oup.com/bioinformatics/article/33/1/49/2525680
10. Deaton, A.M., Bird, A.: CpG islands and the regulation of transcription. Genes & Development 25(10), 1010–1022 (May 2011), http://genesdev.cshlp.org/content/25/10/1010
11. Ecsedi, S., Rodríguez-Aguilera, J.R., Hernandez-Vargas, H.: 5-Hydroxymethylcytosine (5hmc), or How to Identify Your Favorite Cell. Epigenomes 2(1), 3 (Mar 2018), https://www.mdpi.com/2075-4655/2/1/3
12. Ewing, B., Green, P.: Base-Calling of Automated Sequencer Traces Using Phred. II. Error Probabilities. Genome Research 8(3), 186–194 (Mar 1998), http://genome.cshlp.org/content/8/3/186
13. Ewing, B., Hillier, L., Wendl, M.C., Green, P.: Base-Calling of Automated Sequencer Traces UsingPhred. I. Accuracy Assessment. Genome Research 8(3), 175–185 (Mar 1998), http://genome.cshlp.org/content/8/3/175
14. Gers, F.A., Schmidhuber, J., Cummins, F.: Learning to Forget: Continual Prediction with LSTM. Neural Computation 12(10), 2451–2471 (Oct 2000), http://www.mitpressjournals.org/doi/10.1162/089976600300015015
15. Graves, A., Fernández, S., Gomez, F., Schmidhuber, J.: Connectionist Temporal Classification: Labelling Unsegmented Sequence Data with Recurrent Neural Networks. In: Proceedings of the 23rd International Conference on Machine Learning. pp. 369–376. ICML '06, ACM, New York, NY, USA (2006), http://doi.acm.org/10.1145/1143844.1143891, event-place: Pittsburgh, Pennsylvania, USA
16. Graves, A., Jaitly, N.: Towards End-to-End Speech Recognitionwith Recurrent Neural Networks. In: International conference on machine learning. p. 9 (2014)
17. Herceg, Z., Ghantous, A., Wild, C.P., Sklias, A., Casati, L., Duthie, S.J., Fry, R., Issa, J.P., Kellermayer, R., Koturbash, I., Kondo, Y., Lepeule, J., Lima, S.C.S., Marsit, C.J., Rakyan, V., Saffery, R., Taylor, J.A., Teschendorff, A.E., Ushijima, T., Vineis, P., Walker, C.L., Waterland, R.A., Wiemels, J., Ambatipudi, S., Esposti, D.D., Hernandez-Vargas, H.: Roadmap for investigating epigenome deregulation and environmental origins of cancer. International Journal of Cancer 142(5), 874–882 (2018), https://onlinelibrary.wiley.com/doi/abs/10.1002/ijc.31014
18. Hlady, R.A., Sathyanarayan, A., Thompson, J.J., Zhou, D., Wu, Q., Pham, K., Lee, J.H., Liu, C., Robertson, K.D.: Integrating the Epigenome to Identify Drivers of Hepatocellular Carcinoma. Hepatology 69(2), 639–652 (2019), https://aasldpubs.onlinelibrary.wiley.com/doi/abs/10.1002/hep.30211
19. Hochreiter, S., Schmidhuber, J.: Long Short-Term Memory. Neural Computation 9(8), 1735–1780 (Nov 1997), https://doi.org/10.1162/neco.1997.9.8.1735

20. Ioffe, S., Szegedy, C.: Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. arXiv:1502.03167 [cs] (Feb 2015), http://arxiv.org/abs/1502.03167, arXiv: 1502.03167
21. Jabbari, K., Bernardi, G.: Cytosine methylation and CpG, TpG (CpA) and TpA frequencies. Gene 333, 143–149 (May 2004), https://linkinghub.elsevier.com/retrieve/pii/S0378111904000836
22. Jain, M., Olsen, H.E., Paten, B., Akeson, M.: The Oxford Nanopore MinION: delivery of nanopore sequencing to the genomics community. Genome Biology 17(1), 239 (Nov 2016), https://doi.org/10.1186/s13059-016-1103-0
23. Jones, P.A.: Functions of DNA methylation: islands, start sites, gene bodies and beyond. Nature Reviews Genetics 13(7), 484–492 (Jul 2012), http://www.nature.com/articles/nrg3230
24. Kasianowicz, J.J., Brandin, E., Branton, D., Deamer, D.W.: Characterization of individual polynucleotide molecules using a membrane channel. Proceedings of the National Academy of Sciences 93(24), 13770–13773 (Nov 1996), https://www.pnas.org/content/93/24/13770
25. Kress, W.J., Erickson, D.L.: DNA barcodes: Genes, genomics, and bioinformatics. Proceedings of the National Academy of Sciences of the United States of America 105(8), 2761–2762 (Feb 2008), https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2268532/
26. Laszlo, A.H., Derrington, I.M., Brinkerhoff, H., Langford, K.W., Nova, I.C., Samson, J.M., Bartlett, J.J., Pavlenok, M., Gundlach, J.H.: Detection and mapping of 5-methylcytosine and 5-hydroxymethylcytosine with nanopore MspA. Proceedings of the National Academy of Sciences 110(47), 18904–18909 (Nov 2013), http://www.pnas.org/cgi/doi/10.1073/pnas.1310240110
27. Ma, Q., Xu, Z., Lu, H., Xu, Z., Zhou, Y., Yuan, B., Ci, W.: Distal regulatory elements identified by methylation and hydroxymethylation haplotype blocks from mouse brain. Epigenetics & Chromatin 11 (Dec 2018), https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6311040/
28. Rand, A.C., Jain, M., Eizenga, J.M., Musselman-Brown, A., Olsen, H.E., Akeson, M., Paten, B.: Mapping DNA methylation with high-throughput nanopore sequencing. Nature Methods 14(4), 411–413 (Apr 2017), https://www.nature.com/articles/nmeth.4189
29. Rang, F.J., Kloosterman, W.P., de Ridder, J.: From squiggle to basepair: computational approaches for improving nanopore sequencing read accuracy. Genome Biology 19(1), 90 (Jul 2018), https://doi.org/10.1186/s13059-018-1462-9
30. Saxonov, S., Berg, P., Brutlag, D.L.: A genome-wide analysis of CpG dinucleotides in the human genome distinguishes two distinct classes of promoters. Proceedings of the National Academy of Sciences 103(5), 1412–1417 (Jan 2006), https://www.pnas.org/content/103/5/1412
31. Schuster, M., Paliwal, K.K.: Bidirectional recurrent neural networks. IEEE Transactions on Signal Processing 45(11), 2673–2681 (Nov 1997)
32. Simpson, J.T., Workman, R.E., Zuzarte, P.C., David, M., Dursi, L.J., Timp, W.: Detecting DNA cytosine methylation using nanopore sequencing. Nature Methods 14(4), 407–410 (Apr 2017), https://www.nature.com/articles/nmeth.4184
33. Sood, A.J., Viner, C., Hoffman, M.M.: DNAmod: the DNA modification database. Journal of Cheminformatics 11(1), 30 (Apr 2019), https://doi.org/10.1186/s13321-019-0349-4
34. Stoiber, M., Brown, J.: BasecRAWller: Streaming Nanopore Basecalling Directly from Raw Signal. bioRxiv p. 133058 (May 2017), https://www.biorxiv.org/content/10.1101/133058v1

35. Teng, H., Cao, M.D., Hall, M.B., Duarte, T., Wang, S., Coin, L.J.M.: Chiron: translating nanopore raw signal directly into nucleotide sequence using deep learning. GigaScience 7(5) (May 2018), https://academic.oup.com/gigascience/article/7/5/giy037/4966989
36. Vesely, M.D., Kershaw, M.H., Schreiber, R.D., Smyth, M.J.: Natural Innate and Adaptive Immunity to Cancer. Annual Review of Immunology 29(1), 235–271 (Apr 2011), http://www.annualreviews.org/doi/10.1146/annurev-immunol-031210-101324
37. Wick, R.R., Judd, L.M., Holt, K.E.: Performance of neural network basecalling tools for Oxford Nanopore sequencing. bioRxiv p. 543439 (Feb 2019), https://www.biorxiv.org/content/10.1101/543439v1
38. Ziegler, S.F., Buckner, J.H.: FOXP3 and the Regulation of Treg/Th17 Differentiation. Microbes and infection / Institut Pasteur 11(5), 594–598 (Apr 2009), https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2728495/

Annexe

.1 Processus cellulaires dérégulés

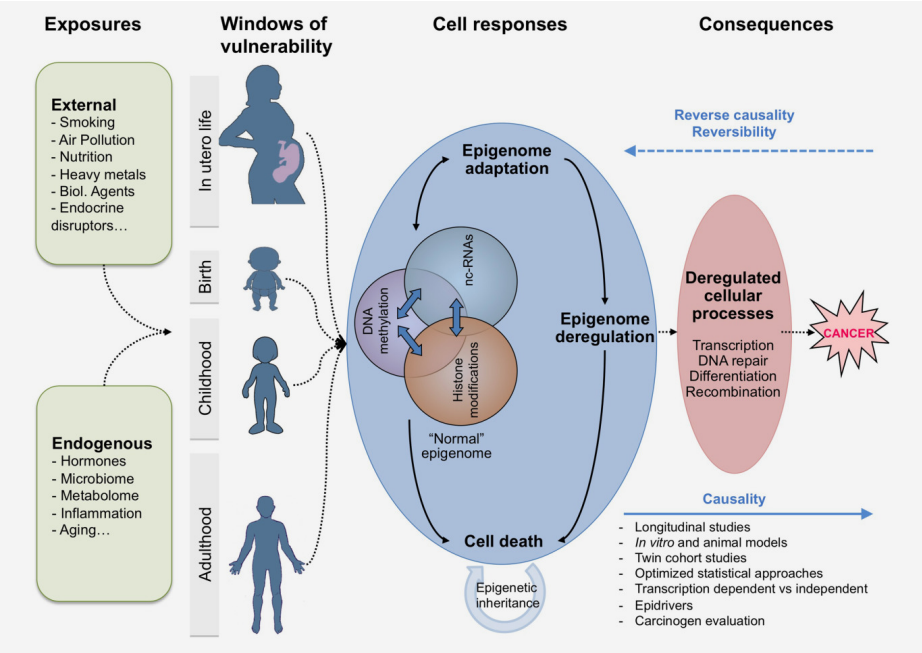

Fig. 17: L'exposition à des sources extérieures ainsi que des processus internes peuvent induire des changements stables et potentiellement réversibles de l'épigénome. Les "signatures" et la persistance de ces altérations dépendent de multiples facteurs comme la durée de l'exposition, le type de tissu et l'étape de développement (Les périodes de la puberté ou de la vie intra-utérine peuvent être particulièrement sensibles). (extrait de [17])

.2 Distribution de la longueur des séquences de l'ADN standard complètement non méthylé

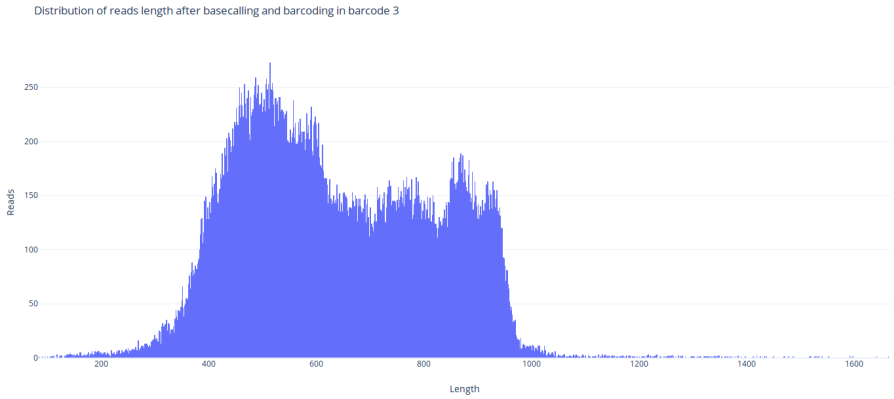

Fig. 18: Les séquences avec le code barre 3 font partie de l'échantillon non méthylé. Nous savons que les séquences font 900 bases, cependant on observe après *basecalling* une distribution de longueur majoritairement entre 350 et 900, avec deux modes à 500 et 900. Durant la préparation de l'ADN, les manipulations peuvent l'endommager, parfois briser le brin, ceci explique la présence d'un mode entre 450 et 500, les brin d'ADN peuvent être coupé en deux ce qui donne un grand nombre de séquences de longueur 450. Les séquences plus longues peuvent être des contaminations, l'ADN d'une bactérie qui s'est glissée dans l'échantillon par exemple ou bien des erreurs de prédiction du *basecaller*. La grande variation de tailles de séquences peut être attribuée à des erreurs de prédiction également.