UNIVERSITÉ DE MONTPELLIER
PLACE EUGÈNE BATAILLON 34095 MONTPELLIER CEDEX 5
Master Bioinformatique – 2023/2024

Développement opérationnel avancé
Travaux Pratiques n° 2

Réalisation d'une API de manipulation de séquences biologiques au formats FASTA & FASTQ

Alban MANCHERON

1 Description du format FASTQ

Le format FASTQ permet de représenter les séquences nucléiques (typiquement issues de séquenceurs à haut-débit) avec les données de qualité initialement produites par le logiciel *Phred*. Ce format est essentiellement utilisé par le *Sanger Institute* et sa structure a été reprise par *Solexa/Illumina* (la différence est sémantique et concerne le calcul des valeurs de qualité).

Ce format est inspiré du format FASTA (*cf.* TP n° 1) et reprend donc l'idée de débuter une séquence par un caractère spécifique : le caractère '`@`'. Cette ligne est suivie par la séquence biologique décrite selon le standard IUPAC (sur une ou plusieurs lignes consécutives). La fin de la séquence est dénotée par une ligne débutant par le symbole '+' et suivie en option par une redite du descriptif de la séquence (si ce descriptif est optionnel, il doit toutefois – s'il est spécifié – être identique au descriptif suivant le symbole '`@`'). Se trouvent ensuite les informations de qualité codées par des symboles ASCII comprises entre '`!`' et '~' (codes allant de 33 à 126 – soit les 94 premiers symboles visibles après l'espace '␣'). L'information de qualité étant spécifique à chaque acide nucléique de la séquence biologique, les deux séquences (biologique et qualité) sont nécessairement de même longueur (hors espaces et sauts de ligne) et le $i^{ème}$ symbole de la séquence qualité se réfère au $i^{ème}$ acide nucléique de la séquence biologique. La fin de la séquence qualité se termine nécessairement par un retour à la ligne et est atteinte lorsque, à chaque symbole de la séquence biologique a été correctement assignée une valeur de qualité.

Exemple de séquence au format FASTQ :

```
@HWUSI-EAS454_0006:1:1:1264:18152#ACAGTG/1
TGTCAAGTGGCATTCTCCCTGCATTCACGGAGGGTATTGCTCGGTCTGTAAATGTTGTGCATGTTGTTGACACTAC
+HWUSI-EAS454_0006:1:1:1264:18152#ACAGTG/1
cc^cc^\‘cce‘ceefeefedd\eeeeea\‘ccLY]_\Xabcb‘_edecd\adY‘‘a‘^^aa\a\^ad‘\T‘TZ‘^
```

De même que la ligne descriptive du format FASTA est généralement une succession de valeurs séparées par des '|', la ligne descriptive d'une séquence au format FASTQ est généralement une ligne de valeurs séparés par des ':'.

Selon la technologie utilisée, les valeurs de type « *Phred* » sont calculées différemment. Initialement, il s'agit d'une simple fonction logarithmique de la probabilité d'erreur de la méthode de « *base-calling* » utilisée, discrétisée et translatée dans l'intervalle $[33; 126]$; les valeurs de score qui ne seraient pas comprises entre 0 et 93 après arrondi sont tronquées à ces bornes, puis sont codées en ajoutant 33 pour obtenir un code ASCII visible (*e.g.*, le code correspondant au caractère '`@`' – et qui peut donc apparaître ailleurs qu'en début de bloc – étant 64, ce symbole représente donc un score de qualité de 31, signifiant dans ce modèle que la base associée a une probabilité d'être fausse inférieure à 10^{-3}). Concernant d'autres technologies telles *Solexa/Illumina*, le calcul du score et son encodage dépend de la version logicielle de leur outil *Illumina Genome Analyzer*. Depuis la version 1.5, le mode de calcul s'est rapproché de celui utilisé par le *Sanger Institute*.

Vous trouverez de plus amples informations sur le site de Wikipedia [1], sur le site du NCBI [2], ou encore sur le site de l'outil de *mapping* MAQ [3].

2 Objectif du TP

Votre travail consiste à compléter l'API (*Application Programming Interface*) du TP 1 afin de permettre la manipulation des fichiers au format FASTQ en plus de celle des fichiers au format FASTA.

Il vous faudra ensuite permettre de « nettoyer » les séquences (correspondant généralement à des *reads* provenant de séquençages à haut débit), d'une part en permettant de supprimer un préfixe donné des séquences lorsqu'il apparaît (utile pour supprimer les *linkers* éventuels en début de chaque *read*), et d'autre part de supprimer les séquences ou portions de séquences non significatives (*e.g.*, queues poly-`A`/poly-`T`, séquences ayant une mauvaise valeur de qualité globale, alphabet dégénéré, ...).

Avant d'implémenter la classe, il vous est largement suggéré de modifier votre API de manipulation des fichiers au format FASTA pour intégrer des concepts objets (héritage, interfaces, polymorphisme, ...)

Votre API devra donc permettre de lire un fichier au format FASTA ou FASTQ (avec et sans détection automatique), vérifier sa conformité (en affichant au besoin les recommandations non respectées et les erreurs rencontrées afin d'assister un éventuel utilisateur dans la correction de ses fichiers).

Les fonctionnalités implémentées dans le précédent TP devront être maintenues et adaptées (comptage du nombre de séquences présentes dans le fichier, analyse des entêtes des séquences, calcul des séquences complémentaires inversées, extraction de sous-séquences, ...).

Vous veillerez comme précédemment à vérifier que les solutions que vous proposerez se comportent en pratique comme attendu en théorie [4].

Pour tester votre API, vous pouvez récupérer un fichier de test sur l'espace *moodle* de l'UE.

Bon Courage...

1. http://en.wikipedia.org/
2. http://www.ncbi.nlm.nih.gov/
3. http://maq.sourceforge.net/fastq.shtml
4. En C++ (sous GNU/Linux), vous pourrez utiliser `getrusage(int, struct rusage *)`.