Lecture de séquences ADN: *K*-mers et *K*-mers espacés

Gabriel Toublanc
Master IGIS ITA
2015 - 2017

Introduction

Les séquenceurs ADN ont connu un développement important depuis le milieu des années 2000, ce qui a permis à la bioinformatique de développer de nouvelles techniques d'analyse.

Le séquençage ADN est constitué de deux principales familles de lectures :

⟶ **Les lectures courtes** (Illumina, Roche, ...)
⟶ **Les lectures longues** (Nanopore, PacBio, ...)

Les **lectures courtes** ont le net avantage d'être **très précises**, avec un taux d'erreur de moins de 1 %, la plupart étant des erreurs de substitution. Leur principal inconvénient est que **de nombreuses lectures** courtes sont nécessaires afin de reconstruire le génome, ce qui est coûteux et peu pratique.

Les **lectures longues**, en revanche, ont l'avantage de ne nécessiter que d'**une petite quantité** d'entre elles pour couvrir le génome d'un être vivant, mais le gros désavantage d'être **hautement imprécises** (15 à 30 % d'erreurs), principalement des erreurs d'insertions et de délétion.

Dans les faits, le génome peut être reconstruit grâce à un nombre déraisonnable de lectures courtes, qui produiront un génome trop fragmenté, ou en assemblant les lectures longues corrigées par les lectures courtes. Plusieurs **méthodes de corrections** ont été développées à ce sujet [1].

L'étude des ***k*-mers** des lectures longues a pour but de pouvoir corriger les lectures longues par elles-mêmes, et ainsi de pouvoir se passer des lectures courtes.

K-mers

Un ***k*-mer** est un facteur de taille **k** d'une séquence ADN donnée. Pour une séquence de taille **L**, on a donc $\mathbf{L} - k + \mathbf{1}$ k-mers possibles.

Ex: Avec la séquence $\mathbf{s} = \mathbf{AACCGGTT}$, on obtient les k-mers de taille 6 (6-mers) suivants:

$\mathbf{k_1}$:	A	A	C	C	G	G	T	T
$\mathbf{k_2}$:	A	A	C	C	G	G	T	T
$\mathbf{k_3}$:	A	A	C	C	G	G	T	T

6-mers: {AACCGG, ACCGGT, CCGGTT}

Afin d'énumérer les k-mers de grands jeux de données, divers outils logiciels ont été développés. L'objectif de ces outils est d'extraire les k-mers, **leur nombre d'occurrences**, d'avoir un **temps d'exécution** viable et d'utiliser le moins d'**espace mémoire** possible.

Le logiciel **Jellyfish** [2] surpasse ses concurrents dans ce domaine, avec des performances bien supérieures, tout facteur confondu.

Les k-mers sont principalement utilisés pour **l'alignement et l'assemblage** de lectures.

Dans le cas présent, l'utilisation des k-mers a pour but de **trouver des répétitions** au sein des lectures longues afin de pouvoir **identifier les nucléotides erronées**, et de les corriger.

K-mers espacés à délétion

Les ***k*-mers espacés** sont utilisés afin de simuler des corrections aux erreurs d'insertions et de délétions sur les lectures longues. Au lieu de prendre les facteurs de taille k d'une séquence, on va sélectionner les k-mers espacés selon un motif précis.

Avec les ***k*-mers espacés à délétion**, chaque **0** du motif correspond à un nucléotide à supprimer.
Par ex, un motif $\mathbf{m} = \mathbf{111011}$ permettra d'extraire tous les 5-mers espacés à partir des 6-mers en supprimant le 4^{eme} nucléotide.

Ex: Avec la séquence $\mathbf{s} = \mathbf{AACCGGTT}$ et le motif $m = 111011$, on obtient les 5-mers espacés suivants:

$\mathbf{k_1}$:	A	A	C	~~C~~	G	G	T	T
$\mathbf{k_2}$:	A	A	C	C	~~G~~	G	T	T
$\mathbf{k_3}$:	A	A	C	C	G	~~G~~	T	T

5-mers: {AACGG, ACCGT, CCGTT}

K-mers espacés à insertion

Avec les ***k*-mers espacés à insertion**, chaque **0** du motif correspond à un nucléotide à insérer.
Tous les nucléotides possibles pour chaque **0** sont ajoutés, ce qui produit $\mathbf{4^t}$ fois plus k-mers, avec **t** =nombre de zéros dans le motif.

Ex: Avec la séquence $\mathbf{s} = \mathbf{AACCGGTT}$ et le motif $\mathbf{m} = \mathbf{111011}$, on obtient les 6-mers espacés suivants:

$\mathbf{k_1}$:	A	A	C	$\overbrace{A,C,G,T}$	C	G	G	T	T
$\mathbf{k_2}$:	A	A	C	C	$\overbrace{A,C,G,T}$	G	G	T	T
$\mathbf{k_3}$:	A	A	C	C	G	$\overbrace{A,C,G,T}$	G	T	T
$\mathbf{k_4}$:	A	A	C	C	G	G	$\overbrace{A,C,G,T}$	T	T

6-mers: { AAC**A**CG, ACC**A**GG, CCG**A**GT, CGG**A**TT,
AAC**C**CG, ACC**C**GG, CCG**C**GT, CGG**C**TT,
AAC**G**CG, ACC**G**GG, CCG**G**GT, CGG**G**TT,
AAC**T**CG, ACC**T**GG, CCG**T**GT, CGG**T**TT }

Références

[1] Pierre Morisse, Thierry Lecroq, and Arnaud Lefebvre. HG-CoLoR: Hybrid-Graph for the error Correction of Long Reads. In *Actes des Journées Ouvertes Biologie Informatique et Mathématiques*, 2017.

[2] Guillaume Marcais and Carl Kingsford. A fast, lock-free approach for efficient parallel counting of occurrences of k-mers. *Bioinformatics*, 27(6):764–770, 2011.