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

Gabriel Toublanc
Master IGIS ITA
2015 - 2017

Introduction

L'analyse de séquençage ADN appliqué à la Bio-Informatique a connu un développement important dans les années 2000.

Le séquençage ADN est effectué de deux façons différentes, par le biais de lectures sur des brins d'ADN:

$\longrightarrow$ **Les lectures longues** (Nanopore, PacBio, ...)

$\longrightarrow$ **Les lectures courtes** (Illumina, Roche, ...)

Les **lectures longues** ont l'avantage de nécessiter que d'**une petite quantité** d'entre elles pour quadriller le génome d'un être vivant, mais le gros désavantage d'être **hautement imprécises** (15 à 30 % d'erreurs)

Les **lectures courtes**, en revanche, ont le net avantage d'être **très précises**, avec un taux d'erreur de moins de 1 %. Leur inconvénient principal est qu'il faut **de nombreuses lectures** courtes afin de reconstruire le génome, ce qui est coûteux et peu pratique.

Dans les faits, le génome peut être reconstruit grâce à un nombre déraisonnable de lectures courtes, ou en corrigeant les lectures longues grâce aux lectures courtes. Plusieurs **méthodes de corrections** ont été développées à ce sujet [1].

Afin de pouvoir se passer des lectures courtes, et d'ainsi corriger les lectures longues d'elles-mêmes, une branche de la bioinformatique s'intéresse à l'étude des **k-mers**.

K-mers

Les **k-mers** sont les facteurs de taille **k** d'une séquence ADN donnée. Pour une lecture de taille **L**, on a donc $\mathbf{L - k + 1}$ k-mers possibles.

Ex: Avec une séquence $x = AACCGGTT$, on a 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 de façon à **grouper** les doublons, 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 meilleures performances bien supérieures, tout facteur confondu.

Les k-mers ainsi extraits 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**, afin de pouvoir les corriger.

K-mers espacés à délétion

Les **k-mers espacés** sont utilisés afin de simuler les 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 une nucléotide à supprimer.
Par ex, un motif $\mathbf{m = 111011}$ créera tous les 5-mers espacés à partir des 6-mers en supprimant le 4^{eme} nucléotide.

Ex: Avec une séquence $x = AACCGGTT$ et le motif $m = 111011$, on a les 5-mers espacés suivants:

$\mathbf{k_1}$: $A\ A\ C\ \not{C}\ G\ G\ T\ T$

$\mathbf{k_2}$: $A\ A\ C\ C\ \not{G}\ G\ T\ T$

$\mathbf{k_3}$: $A\ A\ C\ C\ G\ \not{G}\ T\ T$

5-mers: {AACGG, ACCGT, CCGTT}

K-mers espacés à insertion

Avec les **k-mers espacés à insertion**, chaque **0** du motif correspond à une nucléotide à insérer.
Toutes les nucléotides possibles pour chaque **0** sont extraites, ce qui produit $(\mathbf{L - k + 1 + t}) * \mathbf{4^t}$ k-mers espacés possibles, avec $\mathbf{t}$ = nombre de zéros dans le motif.

Ex: Avec une séquence $\mathbf{x = AACCGGTT}$ et le motif $\mathbf{m = 111011}$, on a 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, AAC**C**CG, AAC**G**CG, AAC**T**CG, ACC**A**GG, ACC**C**GG, ACC**G**GG, ACC**T**GG, CCG**A**GT, CCG**C**GT, CCG**G**GT, CCG**T**GT, CGG**A**TT, CGG**C**TT, CGG**G**TT, CGG**T**TT }

Références

References

[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.