Memoire de stage de Maîtrise IGIS ITA

K-mers et k-mers espacés

Gabriel Toublanc

26 janvier 2017

Université de Rouen, U.F.R des Sciences et Techniques de Saint-Etienne-du-Rouvray, Equipe LITIS TIBS

Introduction

Introduction

Comptage des k-mers

K-mers

Les k-mers sont des facteurs de séquences ADN.

K-mers

Les k-mers sont des facteurs de séquences ADN.

Ex : Avec une séquence $x = AACCGGTT$, on a les k-mers de taille 6 (6-mers) suivants :

K-mers

Les k-mers sont des facteurs de séquences ADN.

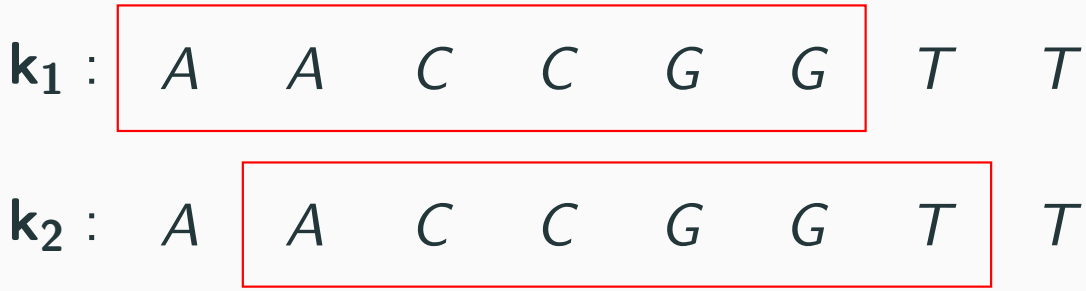
Ex : Avec une séquence $x = AACCGGTT$, on a les k-mers de taille 6 (6-mers) suivants :

$$\mathbf{k_1} : \boxed{A \quad A \quad C \quad C \quad G \quad G} \quad T \quad T$$

K-mers

Les k-mers sont des facteurs de séquences ADN.

Ex : Avec une séquence $x = AACCGGTT$, on a les k-mers de taille 6 (6-mers) suivants :



K-mers

Les k-mers sont des facteurs de séquences ADN.

Ex : Avec une séquence $x = AACCGGTT$, on a les k-mers de taille 6 (6-mers) suivants :

$$\mathbf{k_1}: \boxed{A\ A\ C\ C\ G\ G}\ T\ T$$

$$\mathbf{k_2}: A\ \boxed{A\ C\ C\ G\ G\ T}\ T$$

$$\mathbf{k_3}: A\ A\ \boxed{C\ C\ G\ G\ T\ T}$$

K-mers espacés à délétion

Utilisé afin de simuler les erreurs d'insertions sur les lectures longues.

K-mers espacés à délétion

Utilisé afin de simuler les erreurs d'insertions sur les lectures longues.

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

K-mers espacés à délétion

Utilisé afin de simuler les erreurs d'insertions sur les lectures longues.

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

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

K-mers espacés à délétion

Utilisé afin de simuler les erreurs d'insertions sur les lectures longues.

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

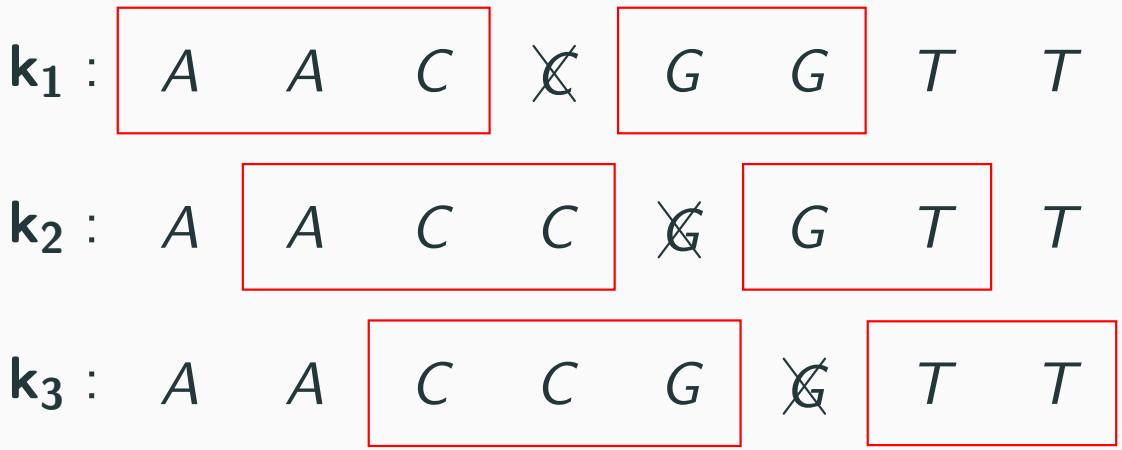
$$\mathbf{k_1} : \boxed{A \quad A \quad C} \not{C} \boxed{G \quad G} \quad T \quad T$$

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

K-mers espacés à délétion

Utilisé afin de simuler les erreurs d'insertions sur les lectures longues.

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



K-mers espacés à insertion

Utilisé afin de simuler les erreurs de délétion sur les lectures longues.

K-mers espacés à insertion

Utilisé afin de simuler les erreurs de délétion sur les lectures longues.

$(\mathbf{L} - \mathbf{k} + \mathbf{1}) * \mathbf{4^t}$ k-mers espacés à insertion possibles.

K-mers espacés à insertion

Utilisé afin de simuler les erreurs de délétion sur les lectures longues.

$(\mathbf{L} - \mathbf{k} + \mathbf{1}) * \mathbf{4^t}$ k-mers espacés à insertion possibles.

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

K-mers espacés à insertion

Utilisé afin de simuler les erreurs de délétion sur les lectures longues.

$(\mathbf{L} - \mathbf{k} + \mathbf{1}) * \mathbf{4^t}$ k-mers espacés à insertion possibles.

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

$$\mathbf{k_1} : \boxed{A \quad A \quad C \overbrace{\quad}^{A,\,C,\,G,\,T} C \quad G} \quad G \quad T \quad T$$

K-mers espacés à insertion

Utilisé afin de simuler les erreurs de délétion sur les lectures longues.

$(\mathbf{L} - \mathbf{k} + \mathbf{1}) * \mathbf{4^t}$ k-mers espacés à insertion possibles.

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

$$\mathbf{k_1} : \boxed{A \quad A \quad C \overbrace{\quad}^{A,\,C,\,G,\,T} C \quad G} \quad G \quad T \quad T$$

$$\mathbf{k_2} : \quad A \quad \boxed{A \quad C \quad C \overbrace{\quad}^{A,\,C,\,G,\,T} G \quad G} \quad T \quad T$$

K-mers espacés à insertion

Utilisé afin de simuler les erreurs de délétion sur les lectures longues.

$(\mathbf{L} - \mathbf{k} + \mathbf{1}) * \mathbf{4^t}$ k-mers espacés à insertion possibles.

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

$$\mathbf{k_1}: \quad A \quad A \quad C \overbrace{\quad}^{A,\,C,\,G,\,T} \quad C \quad G \quad G \quad T \quad T$$

$$\mathbf{k_2}: \quad A \quad A \quad C \quad C \overbrace{\quad}^{A,\,C,\,G,\,T} \quad G \quad G \quad T \quad T$$

$$\mathbf{k_3}: \quad A \quad A \quad C \quad C \quad G \overbrace{\quad}^{A,\,C,\,G,\,T} \quad G \quad T \quad T$$

K-mers espacés à insertion

Utilisé afin de simuler les erreurs de délétion sur les lectures longues.

$(\mathbf{L} - \mathbf{k} + \mathbf{1}) * \mathbf{4^t}$ k-mers espacés à insertion possibles.

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

$$\mathbf{k_1} : A \quad A \quad C \overbrace{A, C, G, T} \quad C \quad G \quad G \quad T \quad T$$

$$\mathbf{k_2} : A \quad A \quad C \quad C \overbrace{A, C, G, T} \quad G \quad G \quad T \quad T$$

$$\mathbf{k_3} : A \quad A \quad C \quad C \quad G \overbrace{A, C, G, T} \quad G \quad T \quad T$$

$$\mathbf{k_4} : A \quad A \quad C \quad C \quad G \quad G \overbrace{A, C, G, T} \quad T \quad T$$

KmersDel

KmersDel

Entrées : table_hachage *table*, chaînes *lectures*, chaîne *motif*, entier *k*

KmersDel

Entrées : table_hachage *table*, chaînes *lectures*, chaîne *motif*, entier *k*

pour chaque *lecture de lectures* **faire**

KmersDel

Entrées : table_hachage *table*, chaînes *lectures*, chaîne *motif*, entier k

pour chaque *lecture de lectures* **faire**

    pour $i = 0;\ i + k \leq |lecture|;\ i+ = 1$ **faire**

        $kmerEntier = 0$;

        $kmer =$ "" ;

KmersDel

Entrées : table_hachage *table*, chaînes *lectures*, chaîne *motif*, entier k

pour chaque *lecture de lectures* **faire**

    pour $i = 0;\ i + k \leq |lecture|;\ i+ = 1$ **faire**

        $kmerEntier = 0$;

        $kmer = ""$;

        pour $j = 0;\ j < k;\ j+ = 1$ **faire**

            si $motif[j] \neq 0$ **alors** $kmer = kmer + lecture[i + j]$;

        fin

KmersDel

Entrées : table_hachage *table*, chaînes *lectures*, chaîne *motif*, entier k

pour chaque *lecture de lectures* **faire**

- **pour** $i = 0;\ i + k \leq |lecture|;\ i+ = 1$ **faire**
  - $kmerEntier = 0$;
  - $kmer = ""$;
  - **pour** $j = 0;\ j < k;\ j+ = 1$ **faire**
    - **si** $motif[j] \neq 0$ **alors** $kmer = kmer + lecture[i + j]$;
  - **fin**
  - **pour chaque** *nucleotide de kmer* **faire**
    - $kmerEntier* = 4$;
    - **suivant** *valeur de nucleotide* **faire**
      - **cas où** A **faire** ;
      - **cas où** C **faire** $kmerEntier+ = 1$;
      - **cas où** G **faire** $kmerEntier+ = 2$;
      - **cas où** T **faire** $kmerEntier+ = 3$;
    - **fin**
  - **fin**

KmersDel

```
Entrées : table_hachage table, chaînes lectures, chaîne motif, entier k
pour chaque lecture de lectures faire
    pour i = 0; i + k ≤ |lecture|; i+ = 1 faire
        kmerEntier = 0 ;
        kmer = "" ;
        pour j = 0; j < k; j+ = 1 faire
            si motif[j] ≠ 0 alors kmer = kmer + lecture[i + j] ;
        fin
        pour chaque nucleotide de kmer faire
            kmerEntier∗ = 4 ;
            suivant valeur de nucleotide faire
                cas où A faire ;
                cas où C faire kmerEntier+ = 1 ;
                cas où G faire kmerEntier+ = 2 ;
                cas où T faire kmerEntier+ = 3 ;
            fin
        fin
        table[kmerEntier]+ = 1 ;
    fin
fin
```

KmersExpand

KmersExpand

kmerExpand

KmersExpand

kmerExpand

Entrées : chaînes *lectures*, chaîne *motif*, entier k

KmersExpand

kmerExpand

Entrées : chaînes *lectures*, chaîne *motif*, entier *k*

pour chaque *lecture de lectures* **faire**

KmersExpand

kmerExpand

Entrées : chaînes *lectures*, chaîne *motif*, entier k

pour chaque *lecture de lectures* **faire**

    pour $i = 0;\ i + k \leq |lecture|;\ i+ = 1$ **faire**

        $kmer = lecture[i : k]$ $kmerExpandRec(kmer, motif, 0)$

    fin

fin

KmersExpand

kmerExpand

Entrées : chaînes *lectures*, chaîne *motif*, entier k

pour chaque *lecture de lectures* **faire**

    pour $i = 0;\ i + k \leq |lecture|;\ i+ = 1$ **faire**

        $kmer = lecture[i : k]$ $kmerExpandRec(kmer, motif, 0)$

    fin

fin

kmerExpandRec

Entrées : chaîne *kmer*, chaîne *nvKmer*, chaîne *motif*, entier *posMotif*, entier *posKmer*

KmersExpand

kmerExpand

Entrées : chaînes *lectures*, chaîne *motif*, entier k

pour chaque *lecture de lectures* **faire**

    pour $i = 0;\ i + k \leq |lecture|;\ i+ = 1$ **faire**
        $kmer = lecture[i : k]$ $kmerExpandRec(kmer, motif, 0)$
    fin

fin

kmerExpandRec

Entrées : chaîne *kmer*, chaîne *nvKmer*, chaîne *motif*, entier *posMotif*, entier *posKmer*

si $posSeed == |motif|$ **alors** $affiche(nvKmer)$;

KmersExpand

kmerExpand

Entrées : chaînes *lectures*, chaîne *motif*, entier k

pour chaque *lecture de lectures* **faire**

  pour $i = 0;\ i + k \leq |lecture|;\ i+ = 1$ **faire**

    $kmer = lecture[i : k]$ $kmerExpandRec(kmer, motif, 0)$

  fin

fin

kmerExpandRec

Entrées : chaîne *kmer*, chaîne *nvKmer*, chaîne *motif*, entier *posMotif*, entier *posKmer*

si $posSeed == |motif|$ **alors** $affiche(nvKmer)$;

sinon si $motif[posMotif] \neq 0$ **alors**

  $kmersExpandRec(kmer, nvKmer + kmer[posKmer], posMotif + 1, posKmer + 1)$;

KmersExpand

kmerExpand

Entrées : chaînes *lectures*, chaîne *motif*, entier k

pour chaque *lecture de lectures* **faire**

    pour $i = 0;\ i + k \leq |lecture|;\ i+ = 1$ **faire**

        $kmer = lecture[i : k]$ $kmerExpandRec(kmer, motif, 0)$

    fin

fin

kmerExpandRec

Entrées : chaîne *kmer*, chaîne *nvKmer*, chaîne *motif*, entier *posMotif*, entier *posKmer*

si $posSeed == |motif|$ **alors** $affiche(nvKmer)$;

sinon si $motif[posMotif] \neq 0$ **alors**

  $kmersExpandRec(kmer, nvKmer + kmer[posKmer], posMotif + 1, posKmer + 1)$;

sinon

    $kmersExpandRec(kmer, nvKmer + A, posMotif + 1, posKmer)$;

    $kmersExpandRec(kmer, nvKmer + C, posMotif + 1, posKmer)$;

    $kmersExpandRec(kmer, nvKmer + G, posMotif + 1, posKmer)$;

    $kmersExpandRec(kmer, nvKmer + T, posMotif + 1, posKmer)$;

fin

Mots Minimaux absents (MMA)

Mots Minimaux absents (MMA)

Plus long sous-mot commun (PLS)

Plus long sous-mot commun (PLS)

Résultats obtenus

Résultats : KmersDel et kmersExpand

Résultats : Mots Minimaux absents (MMA)

Résultats : Plus long sous-mot commun (PLS)

Conclusion