



HLIN608 – Algorithmique du
texte.

Alban MANCHERON

Titre & Sommaire

Algorithmique du texte

HLIN608

Licence 3 Informatique

Alban MANCHERON

alban.mancheron@lirmm.fr

<http://www.mancheron.infos.st/>

Faculté des Sciences & Techniques de l'Université de Montpellier

Année universitaire 2017–2018, 2^e semestre
[version 1.7]



Sommaire

HLIN608 – Algorithmique du texte.

Alban MANCHERON

Titre & Sommaire

I Présentation (v. 1.7)

II Préambule (v. 1.4)

III Dot-plots et extraction de motifs (v. 1.9)

I – Présentation (v. 1.7)

Sommaire



HLIN608 – Algorithmique du texte.

Alban MANCHERON

Licence 3 Informatique

Organisation du module

I.1/ Licence 3 Informatique

I.2/ Organisation du module



HLIN608 – Algorithmique du texte.

Alban MANCHERON

Licence 3 Informatique

Organisation du module

I.1/ Licence 3 Informatique

I.2/ Organisation du module



Aspects pragmatiques

HLIN608 – Algorithmique du texte.

Alban MANCHERON

Licence 3 Informatique

Organisation du module



Aspects pragmatiques

- outils, logiciels, matériels

HLIN608 – Algorithmique du texte.

Alban MANCHERON

Licence 3 Informatique

Organisation du module



Aspects pragmatiques

- outils, logiciels, matériels
- analyse, manipulation, comparaison à \pm large échelle de données bioinformatiques

HLIN608 – Algorithmique du texte.

Alban MANCHERON

Licence 3 Informatique

Organisation du module



Aspects pragmatiques

- outils, logiciels, matériels
- analyse, manipulation, comparaison à \pm large échelle de données bioinformatiques
- Génomique, Transcriptomique, Protéomique, ...

HLIN608 – Algorithmique du texte.

Alban MANCHERON

Licence 3 Informatique

Organisation du module



Aspects pragmatiques

- outils, logiciels, matériels
 - analyse, manipulation, comparaison à \pm large échelle de données bioinformatiques
 - Génomique, Transcriptomique, Protéomique, ...
- ⋮

HLIN608 – Algorithmique du texte.

Alban MANCHERON

Licence 3 Informatique

Organisation du module



Aspects pragmatiques

- outils, logiciels, matériels
 - analyse, manipulation, comparaison à \pm large échelle de données bioinformatiques
 - Génomique, Transcriptomique, Protéomique, ...
- ⋮

Aspects informatiques

HLIN608 – Algorithmique du texte.

Alban MANCHERON

Licence 3 Informatique

Organisation du module



Aspects pragmatiques

- outils, logiciels, matériels
 - analyse, manipulation, comparaison à \pm large échelle de données bioinformatiques
 - Génomique, Transcriptomique, Protéomique, ...
- ⋮

Aspects informatiques

- structures de données



Aspects pragmatiques

- outils, logiciels, matériels
 - analyse, manipulation, comparaison à \pm large échelle de données bioinformatiques
 - Génomique, Transcriptomique, Protéomique, ...
- ⋮

Aspects informatiques

- structures de données
- algorithmique



Aspects pragmatiques

- outils, logiciels, matériels
 - analyse, manipulation, comparaison à \pm large échelle de données bioinformatiques
 - Génomique, Transcriptomique, Protéomique, ...
- ⋮

Aspects informatiques

- structures de données
 - algorithmique
- ⋮



Aspects pragmatiques

- outils, logiciels, matériels
 - analyse, manipulation, comparaison à \pm large échelle de données bioinformatiques
 - Génomique, Transcriptomique, Protéomique, ...
- ⋮

Aspects informatiques

- structures de données
 - algorithmique
- ⋮

Aspects biologiques



Aspects pragmatiques

- outils, logiciels, matériels
 - analyse, manipulation, comparaison à \pm large échelle de données bioinformatiques
 - Génomique, Transcriptomique, Protéomique, ...
- ⋮

Aspects informatiques

- structures de données
 - algorithmique
- ⋮

Aspects biologiques

- analyse/comparaison de séquences



Aspects pragmatiques

- outils, logiciels, matériels
 - analyse, manipulation, comparaison à \pm large échelle de données bioinformatiques
 - Génomique, Transcriptomique, Protéomique, ...
- ⋮

Aspects informatiques

- structures de données
 - algorithmique
- ⋮

Aspects biologiques

- analyse/comparaison de séquences
- technologies de séquençages



Aspects pragmatiques

- outils, logiciels, matériels
 - analyse, manipulation, comparaison à \pm large échelle de données bioinformatiques
 - Génomique, Transcriptomique, Protéomique, ...
- ⋮

Aspects informatiques

- structures de données
 - algorithmique
- ⋮

Aspects biologiques

- analyse/comparaison de séquences
- technologies de séquençages
- *mapping*



Aspects pragmatiques

- outils, logiciels, matériels
- analyse, manipulation, comparaison à \pm large échelle de données bioinformatiques
- Génomique, Transcriptomique, Protéomique, ...

⋮

Aspects informatiques

- structures de données
- algorithmique

⋮

Aspects biologiques

- analyse/comparaison de séquences
- technologies de séquençages
- *mapping*

⋮

Organisation du module

Sommaire



I.1/ Licence 3 Informatique

I.2/ Organisation du module

- Emploi du temps
- Objectifs

HLIN608 – Algorithmique du texte.

Alban MANCHERON

Licence 3 Informatique

Organisation du module

Emploi du temps

Objectifs

Organisation du module

Emploi du temps



HLIN608 – Algorithmique du texte.

Alban MANCHERON

Licence 3 Informatique

Organisation du module

Emploi du temps

Objectifs



Organisation du module

Emploi du temps



Équipe pédagogique

- Alban MANCHERON [13h30 CM/TD/TP]
⇒ Introduction, DotPlot simplifié,
recherche/extraction/indexation de motifs
- Sylvain DAUDÉ [18h00 CM/TD/TP]
⇒ Bords/périodes, Mots de FIBONACCI, Alignement,
⇒ Sous-séquences communes, DotPlot.
- Annie CHATEAU (resp. UE) [13h30 CM/TD/TP]
Assemblage *de novo*, Graphes et théorie de la
complexité (problèmes NP)



HLIN608 – Algorithmique du texte.

Alban MANCHERON

Licence 3 Informatique

Organisation du module

Emploi du temps

Objectifs

Organisation du module

Emploi du temps



Équipe pédagogique

- Alban MANCHERON [13h30 CM/TD/TP]
⇒ Introduction, DotPlot simplifié,
recherche/extraction/indexation de motifs
- Sylvain DAUDÉ [18h00 CM/TD/TP]
⇒ Bords/périodes, Mots de FIBONACCI, Alignement,
⇒ Sous-séquences communes, DotPlot.
- Annie CHATEAU (resp. UE) [13h30 CM/TD/TP]
Assemblage *de novo*, Graphes et théorie de la
complexité (problèmes NP)



HLIN608 – Algorithmique du texte.

Alban MANCHERON

Licence 3 Informatique

Organisation du module

Emploi du temps

Objectifs

Évaluation (5 ECTS)

- CP : 30% (projet+écrit)
- CT : 70%

Organisation du module

Objectifs



HLIN608 – Algorithmique du texte.

Alban MANCHERON

Licence 3 Informatique

Organisation du module

Emploi du temps

Objectifs

Pour nous [les enseignants]



Organisation du module

Objectifs

Pour nous [les enseignants]

- vous initier à l'algorithmique du texte

HLIN608 – Algorithmique du texte.

Alban MANCHERON

Licence 3 Informatique

Organisation du module

Emploi du temps

Objectifs

Organisation du module

Objectifs



HLIN608 – Algorithmique du texte.

Alban MANCHERON

Licence 3 Informatique

Organisation du module

Emploi du temps

Objectifs

Pour nous [les enseignants]

- vous initier à l’algorithmique du texte
- faire découvrir la bioinformatique

Organisation du module

Objectifs



HLIN608 – Algorithmique du texte.

Alban MANCHERON

Licence 3 Informatique

Organisation du module

Emploi du temps

Objectifs

Pour nous [les enseignants]

- vous initier à l’algorithmique du texte
- faire découvrir la bioinformatique
- faire comprendre la bioinformatique

Organisation du module

Objectifs



HLIN608 – Algorithmique du texte.

Alban MANCHERON

Licence 3 Informatique

Organisation du module

Emploi du temps

Objectifs

Pour nous [les enseignants]

- vous initier à l’algorithmique du texte
- faire découvrir la bioinformatique
- faire comprendre la bioinformatique

Pour vous [les étudiants]

Organisation du module

Objectifs



HLIN608 – Algorithmique du texte.

Alban MANCHERON

Licence 3 Informatique

Organisation du module

Emploi du temps

Objectifs

Pour nous [les enseignants]

- vous initier à l’algorithmique du texte
- faire découvrir la bioinformatique
- faire comprendre la bioinformatique

Pour vous [les étudiants]

- avoir plaisir à faire de l’algorithmique

Organisation du module

Objectifs



HLIN608 – Algorithmique du texte.

Alban MANCHERON

Licence 3 Informatique

Organisation du module

Emploi du temps

Objectifs

Pour nous [les enseignants]

- vous initier à l’algorithmique du texte
- faire découvrir la bioinformatique
- faire comprendre la bioinformatique

Pour vous [les étudiants]

- avoir plaisir à faire de l’algorithmique
- mieux choisir votre orientation

II – Préambule (v. 1.4)

Sommaire



HLIN608 – Algorithmique du texte.

Alban MANCHERON

Rappels théoriques

Rappels de complexité

Bonnes pratiques

II.1/ Rappels théoriques

II.2/ Rappels de complexité

II.3/ Bonnes pratiques



Rappels théoriques

Sommaire

II.1/ Rappels théoriques

- Algorithmique du texte
- Artillerie mathématique
- Ordres de grandeur

II.2/ Rappels de complexité

II.3/ Bonnes pratiques

HLIN608 – Algorithmique du texte.

Alban MANCHERON

Rappels théoriques

Algorithmique du texte

Artillerie mathématique

Ordres de grandeur

Rappels de complexité

Bonnes pratiques



Rappels théoriques

Algorithmique du texte

Vocabulaire

Alphabet $\longrightarrow \Sigma = \{A, C, G, T\}$

HLIN608 – Algorithmique du texte.

Alban MANCHERON

Rappels théoriques

Algorithmique du texte

Artillerie mathématique

Ordres de grandeur

Rappels de complexité

Bonnes pratiques

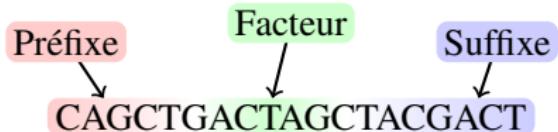


Rappels théoriques

Algorithmique du texte

Vocabulaire

Alphabet $\longrightarrow \Sigma = \{A, C, G, T\}$



HLIN608 – Algorithmique du texte.

Alban MANCHERON

Rappels théoriques

Algorithmique du texte

Artillerie mathématique

Ordres de grandeur

Rappels de complexité

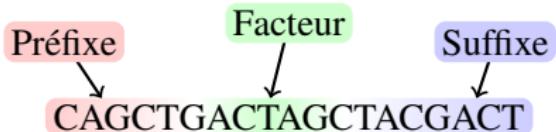
Bonnes pratiques

Rappels théoriques

Algorithmique du texte

Vocabulaire

Alphabet $\longrightarrow \Sigma = \{A, C, G, T\}$



Notations

- Le mot vide est noté ε . Il est de longueur 0
- La longueur du mot $s = \alpha_1 \alpha_2 \cdots \alpha_n$ est égale à $|s| = n$
- Le mot $s[i..j]$ désigne le facteur commençant en i et terminant en j dans s
- On note $s[i]$ le symbole à la position i et $s[i..i] = s[i]$
- Le préfixe de longueur ℓ de s est $s[1..\ell]$
- Le suffixe de longueur ℓ de s est $s[n - \ell + 1..n]$

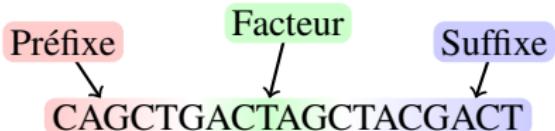
Rappels théoriques

Algorithmique du texte



Vocabulaire

Alphabet $\longrightarrow \Sigma = \{A, C, G, T\}$



Notations

- Le mot vide est noté ε . Il est de longueur 0
- La longueur du mot $s = \alpha_1 \alpha_2 \cdots \alpha_n$ est égale à $|s| = n$
- Le mot $s[i..j]$ désigne le facteur commençant en i et terminant en j dans s
- On note $s[i]$ le symbole à la position i et $s[i..i] = s[i]$
- Le préfixe de longueur ℓ de s est $s[1..\ell]$
- Le suffixe de longueur ℓ de s est $s[n - \ell + 1..n]$

Questions

- Combien de préfixes/suffixes/facteurs pour un mot de longueur n ?
- Combien de sous-séquences pour un mot de longueur n (sachant qu'une sous-séquence s'obtient en enlevant entre 0 et n symboles de la séquence initiale) ?

HLIN608 – Algorithmique du texte.

Alban MANCHERON

Rappels théoriques

Algorithmique du texte

Artillerie mathématique

Ordres de grandeur

Rappels de complexité

Bonnes pratiques

Rappels théoriques

Artillerie mathématique



Notations

- Les sommes de suites de nombres sont notées par le symbole \sum
- Les produits de suites de nombres sont notées par le symbole \prod

HLIN608 – Algorithmique du texte.

Alban MANCHERON

Rappels théoriques

Algorithmique du texte

Artillerie mathématique

Ordres de grandeur

Rappels de complexité

Bonnes pratiques

Rappels théoriques

Artillerie mathématique



Séries « classiques »

$$\sum_{i=m}^n 1 = n - m + 1$$

$$\sum_{i=m}^n i = \frac{(n - m + 1)(m + n)}{2}$$

$$\sum_{i=0}^n i^2 = \frac{n(n+1)(2n+1)}{6}$$

$$\sum_{i=m}^n x^i = \frac{x^m - x^{n+1}}{1-x}$$

$$\sum_{i=0}^n \binom{n}{i} = 2^n = 1 + \sum_{i=0}^{n-1} 2^i$$

$$\sum_{i=0}^n \binom{i}{k} = \binom{n+1}{k+1}$$

$$\sum_{i=0}^n 1 = n + 1$$

$$\sum_{i=0}^n i = \frac{n(n+1)}{2} = \sum_{i=1}^n i$$

$$\sum_{i=0}^n i^3 = \left(\frac{n(n+1)}{2} \right)^2 = \left(\sum_{i=0}^n i \right)^2$$

$$\sum_{i=0}^n x^i = \frac{1 - x^{n+1}}{1-x}$$

$$\sum_{i=1}^n i \binom{n}{i} = n 2^{n-1}$$

$$(a+b)^n = \sum_{i=0}^n \binom{n}{i} a^{n-i} b^i$$

HLIN608 – Algorithmique du texte.

Alban MANCHERON

Rappels théoriques

Algorithmique du texte

Artillerie mathématique

Ondres de grandeur

Rappels de complexité

Bonnes pratiques

Rappels théoriques

Artillerie mathématique



Séries « classiques »

$$\sum_{i=m}^n 1 = n - m + 1$$

$$\sum_{i=m}^n i = \frac{(n - m + 1)(m + n)}{2}$$

$$\sum_{i=0}^n i^2 = \frac{n(n+1)(2n+1)}{6}$$

$$\sum_{i=m}^n x^i = \frac{x^m - x^{n+1}}{1-x}$$

$$\sum_{i=0}^n \binom{n}{i} = 2^n = 1 + \sum_{i=0}^{n-1} 2^i$$

$$\sum_{i=0}^n \binom{i}{k} = \binom{n+1}{k+1}$$

$$\sum_{i=0}^n 1 = n + 1$$

$$\sum_{i=0}^n i = \frac{n(n+1)}{2} = \sum_{i=1}^n i$$

$$\sum_{i=0}^n i^3 = \left(\frac{n(n+1)}{2} \right)^2 = \left(\sum_{i=0}^n i \right)^2$$

$$\sum_{i=0}^n x^i = \frac{1 - x^{n+1}}{1-x}$$

$$\sum_{i=1}^n i \binom{n}{i} = n 2^{n-1}$$

$$(a+b)^n = \sum_{i=0}^n \binom{n}{i} a^{n-i} b^i$$

HLIN608 – Algorithmique du texte.

Alban MANCHERON

Rappels théoriques

Algorithmique du texte

Artillerie mathématique

Ondres de grandeur

Rappels de complexité

Bonnes pratiques

TD : prouver les formules en rouge



Rappels théoriques

Artillerie mathématique

HLIN608 – Algorithmique du texte.

Alban MANCHERON

Propriétés « classiques »

$$\sum_{n=a}^b \alpha f(n) = \alpha \sum_{n=a}^b f(n)$$

$$\sum_{n=a}^b f(n) = \sum_{n=a+m}^{b+m} f(n-m)$$

$$\sum_{n=a}^b (f(n) + g(n)) = \sum_{n=a}^b f(n) + \sum_{n=a}^b g(n)$$

$$\sum_{n=a}^b f(n) = \sum_{n=a}^c f(n) + \sum_{n=c+1}^b f(n)$$

$$\left(\sum_{n=a}^b f(n) \right) \times \left(\sum_{n=c}^d g(n) \right) = \sum_{m=a}^b \sum_{n=c}^d f(m) g(n) = \sum_{m=c}^d \sum_{n=a}^b f(n) g(m)$$

$$\sum_{n=a}^b \log f(n) = \log \left(\prod_{n=a}^b f(n) \right)$$

$$\alpha^{\left(\sum_{n=a}^b f(n)\right)} = \prod_{n=a}^b \alpha^{f(n)}$$

$$\log_b b^n = b^{\log_b n} = n$$

$$\log_a x = \frac{\log_b x}{\log_b a}$$

Rappels théoriques

Algorithmique du texte

Artillerie mathématique

Ondres de grandeur

Rappels de complexité

Bonnes pratiques



Rappels théoriques

Ordres de grandeur [Notations]

Les notations de LANDAU

- $O(g(n)) : \exists k > 0, n_0 \forall n > n_0 |f(n)| \leq k|g(n)|$
 g est asymptotiqu^t une borne supérieur de f à k près

HLIN608 – Algorithmique du texte.

Alban MANCHERON

Rappels théoriques

Algorithmique du texte

Artillerie mathématique

Ordres de grandeur

Notations

Arithmétique

Rappels de complexité

Bonnes pratiques



Rappels théoriques

Ordres de grandeur [Notations]

Les notations de LANDAU

- $O(g(n)) : \exists k > 0, n_0 \forall n > n_0 |f(n)| \leq k|g(n)|$
 g est asymptotiqu^t une borne supérieur de f à k près
- $\Omega(g(n)) : \exists k > 0, n_0 \forall n > n_0 k|g(n)| \leq |f(n)|$
 g est asymptotiqu^t une borne inférieur de f à tout k près

HLIN608 – Algorithmique du texte.

Alban MANCHERON

Rappels théoriques

Algorithmique du texte

Artillerie mathématique

Ordres de grandeur

Notations

Arithmétique

Rappels de complexité

Bonnes pratiques

Rappels théoriques

Ordres de grandeur [Notations]



HLIN608 – Algorithmique du texte.

Alban MANCHERON

Les notations de LANDAU

- $O(g(n)) : \exists k > 0, n_0 \forall n > n_0 |f(n)| \leq k |g(n)|$
 g est asymptotiqu^t une borne supérieur de f à k près
- $\Omega(g(n)) : \exists k > 0, n_0 \forall n > n_0 k |g(n)| \leq |f(n)|$
 g est asymptotiqu^t une borne inférieur de f à tout k près
- $\Theta(g(n)) :$
 $\exists k_1, k_2 > 0, n_0 \forall n > n_0 k_1 |g(n)| < |f(n)| < k_2 |g(n)|$
 g est asymptotiqu^t une borne inf. et sup. de f à k_1, k_2 près

Rappels théoriques

Algorithmique du texte

Artillerie mathématique

Ordres de grandeur

Notations

Arithmétique

Rappels de complexité

Bonnes pratiques



Rappels théoriques

Ordres de grandeur [Notations]

HLIN608 – Algorithmique du texte.

Alban MANCHERON

Les notations de LANDAU

- $O(g(n)) : \exists k > 0, n_0 \forall n > n_0 |f(n)| \leq k|g(n)|$
 g est asymptotiqu^t une borne supérieur de f à k près
- $\Omega(g(n)) : \exists k > 0, n_0 \forall n > n_0 k|g(n)| \leq |f(n)|$
 g est asymptotiqu^t une borne inférieur de f à tout k près
- $\Theta(g(n)) :$
 $\exists k_1, k_2 > 0, n_0 \forall n > n_0 k_1|g(n)| < |f(n)| < k_2|g(n)|$
 g est asymptotiqu^t une borne inf. et sup. de f à k_1, k_2 près
- $o(g(n)) : \forall \varepsilon > 0 \exists n_0 \forall n > n_0 |f(n)| \leq \varepsilon|g(n)|$
 f est asymptotiqu^t négligeable devant g

Rappels théoriques

Algorithmique du texte

Artillerie mathématique

Ordres de grandeur

Notations

Arithmétique

Rappels de complexité

Bonnes pratiques

Rappels théoriques

Ordres de grandeur [Notations]

Les notations de LANDAU

- $O(g(n)) : \exists k > 0, n_0 \forall n > n_0 |f(n)| \leq k |g(n)|$
 g est asymptotiqu^t une borne supérieur de f à k près
- $\Omega(g(n)) : \exists k > 0, n_0 \forall n > n_0 k |g(n)| \leq |f(n)|$
 g est asymptotiqu^t une borne inférieur de f à tout k près
- $\Theta(g(n)) :$
 $\exists k_1, k_2 > 0, n_0 \forall n > n_0 k_1 |g(n)| < |f(n)| < k_2 |g(n)|$
 g est asymptotiqu^t une borne inf. et sup. de f à k_1, k_2 près
- $o(g(n)) : \forall \varepsilon > 0 \exists n_0 \forall n > n_0 |f(n)| \leq \varepsilon |g(n)|$
 f est asymptotiqu^t négligeable devant g
- $\omega(g(n)) : \forall k > 0 \exists n_0 \forall n > n_0 k |g(n)| \leq |f(n)|$
 g est asymptotiqu^t négligeable devant f

Rappels théoriques

Algorithmique du texte

Artillerie mathématique

Ordres de grandeur

Notations

Arithmétique

Rappels de complexité

Bonnes pratiques

Rappels théoriques

Ordres de grandeur [Notations]

Les notations de LANDAU

- $O(g(n)) : \exists k > 0, n_0 \forall n > n_0 |f(n)| \leq k |g(n)|$
 g est asymptotiqu^t une borne supérieur de f à k près
- $\Omega(g(n)) : \exists k > 0, n_0 \forall n > n_0 k |g(n)| \leq |f(n)|$
 g est asymptotiqu^t une borne inférieur de f à tout k près
- $\Theta(g(n)) :$
 $\exists k_1, k_2 > 0, n_0 \forall n > n_0 k_1 |g(n)| < |f(n)| < k_2 |g(n)|$
 g est asymptotiqu^t une borne inf. et sup. de f à k_1, k_2 près
- $o(g(n)) : \forall \varepsilon > 0 \exists n_0 \forall n > n_0 |f(n)| \leq \varepsilon |g(n)|$
 f est asymptotiqu^t négligeable devant g
- $\omega(g(n)) : \forall k > 0 \exists n_0 \forall n > n_0 k |g(n)| \leq |f(n)|$
 g est asymptotiqu^t négligeable devant f
- $\sim g(n) : \forall \varepsilon > 0 \exists n_0 \forall n > n_0 |f(n)/g(n) - 1| < \varepsilon$
 f et g sont asymptotiqu^t égales

Rappels théoriques

Algorithmique du texte

Artillerie mathématique

Ordres de grandeur

Notations

Arithmétique

Rappels de complexité

Bonnes pratiques

Rappels théoriques

Ordres de grandeur [Arithmétique]



HLIN608 – Algorithmique du texte.

Alban MANCHERON

Les notations de LANDAU

- $h(n) = O(g(n))$ et $g(n) = O(f(n))$
- ⇒ $h(n) = O(f(n))$

Rappels théoriques

Algorithmique du texte

Artillerie mathématique

Ordres de grandeur

Notations

Arithmétique

Rappels de complexité

Bonnes pratiques

Rappels théoriques

Ordres de grandeur [Arithmétique]



HLIN608 – Algorithmique du texte.

Alban MANCHERON

Les notations de LANDAU

- $h(n) = O(g(n))$ et $g(n) = O(f(n))$
⇒ $h(n) = O(f(n))$

- $g_1(n) = O(f_1(n))$ et $g_2(n) = O(f_2(n))$
⇒ $g_1(n) + g_2(n) = O(\max(f_1(n), f_2(n)))$

Rappels théoriques

Algorithmique du texte

Artillerie mathématique

Ordres de grandeur

Notations

Arithmétique

Rappels de complexité

Bonnes pratiques

Rappels théoriques

Ordres de grandeur [Arithmétique]



HLIN608 – Algorithmique du texte.

Alban MANCHERON

Les notations de LANDAU

- $h(n) = O(g(n))$ et $g(n) = O(f(n))$
⇒ $h(n) = O(f(n))$
- $g_1(n) = O(f_1(n))$ et $g_2(n) = O(f_2(n))$
⇒ $g_1(n) + g_2(n) = O(\max(f_1(n), f_2(n)))$
- $g_1(n) = O(f_1(n))$ et $g_2(n) = O(f_2(n))$
⇒ $g_1(n) g_2(n) = O(f_1(n)f_2(n))$

Rappels théoriques

Algorithmique du texte

Artillerie mathématique

Ordres de grandeur

Notations

Arithmétique

Rappels de complexité

Bonnes pratiques



Rappels théoriques

Ordres de grandeur [Arithmétique]

HLIN608 – Algorithmique du texte.

Alban MANCHERON

Les notations de LANDAU

- $h(n) = O(g(n))$ et $g(n) = O(f(n))$
⇒ $h(n) = O(f(n))$
- $g_1(n) = O(f_1(n))$ et $g_2(n) = O(f_2(n))$
⇒ $g_1(n) + g_2(n) = O(\max(f_1(n), f_2(n)))$
- $g_1(n) = O(f_1(n))$ et $g_2(n) = O(f_2(n))$
⇒ $g_1(n) g_2(n) = O(f_1(n)f_2(n))$
- $f(n) = \Theta(g(n))$
\Leftrightarrow $f(n) = O(g(n))$ et $g(n) = O(f(n))$

Rappels théoriques

Algorithmique du texte

Artillerie mathématique

Ordres de grandeur

Notations

Arithmétique

Rappels de complexité

Bonnes pratiques

Rappels de complexité

Sommaire



HLIN608 – Algorithmique du texte.

Alban MANCHERON

II.1/ Rappels théoriques

II.2/ Rappels de complexité

- Mesure de la complexité
- Ordres de grandeur de la complexité
- Intérêt de la complexité

II.3/ Bonnes pratiques

Rappels théoriques

Rappels de complexité

Mesure de la complexité

Ordres de grandeur de la complexité

Intérêt de la complexité

Bonnes pratiques

Rappels de complexité

Mesure de la complexité



HLIN608 – Algorithmique du
texte.

Alban MANCHERON

Rappels théoriques

Rappels de complexité

Mesure de la complexité

Ordres de grandeur de la complexité

Intérêt de la complexité

Bonnes pratiques

?



Rappels de complexité

Mesure de la complexité

En temps

HLIN608 – Algorithmique du texte.

Alban MANCHERON

Rappels théoriques

Rappels de complexité

Mesure de la complexité

Ordres de grandeur de la complexité

Intérêt de la complexité

Bonnes pratiques

Rappels de complexité

Mesure de la complexité



En temps

En espace

HLIN608 – Algorithmique du texte.

Alban MANCHERON

Rappels théoriques

Rappels de complexité

Mesure de la complexité

Ordres de grandeur de la complexité

Intérêt de la complexité

Bonnes pratiques

Rappels de complexité

Mesure de la complexité



En temps

- nombre d'opérations effectuées lors du déroulement de l'algorithme

En espace

HLIN608 – Algorithmique du texte.

Alban MANCHERON

Rappels théoriques

Rappels de complexité

Mesure de la complexité

Ordres de grandeur de la complexité

Intérêt de la complexité

Bonnes pratiques

Rappels de complexité

Mesure de la complexité



En temps

- nombre d'opérations effectuées lors du déroulement de l'algorithme

En espace

- mémoire nécessaire pour le stockage des données calculées lors du déroulement de l'algorithme

HLIN608 – Algorithmique du texte.

Alban MANCHERON

Rappels théoriques

Rappels de complexité

Mesure de la complexité

Ordres de grandeur de la complexité

Intérêt de la complexité

Bonnes pratiques

Rappels de complexité

Mesure de la complexité



En temps

- nombre d'opérations effectuées lors du déroulement de l'algorithme

En espace

- mémoire nécessaire pour le stockage des données calculées lors du déroulement de l'algorithme

Dans tous les cas :

HLIN608 – Algorithmique du texte.

Alban MANCHERON

Rappels théoriques

Rappels de complexité

Mesure de la complexité

Ordres de grandeur de la complexité

Intérêt de la complexité

Bonnes pratiques

Rappels de complexité

Mesure de la complexité



En temps

- nombre d'opérations effectuées lors du déroulement de l'algorithme

En espace

- mémoire nécessaire pour le stockage des données calculées lors du déroulement de l'algorithme

Dans tous les cas :

- mesures en fonction de la forme des données en entrée du problème

HLIN608 – Algorithmique du texte.

Alban MANCHERON

Rappels théoriques

Rappels de complexité

Mesure de la complexité

Ordres de grandeur de la complexité

Intérêt de la complexité

Bonnes pratiques

Rappels de complexité

Mesure de la complexité



En temps

- nombre d'opérations effectuées lors du déroulement de l'algorithme

En espace

- mémoire nécessaire pour le stockage des données calculées lors du déroulement de l'algorithme

Dans tous les cas :

- mesures en fonction de la forme des données en entrée du problème
- étude des cas particuliers

HLIN608 – Algorithmique du texte.

Alban MANCHERON

Rappels théoriques

Rappels de complexité

Mesure de la complexité

Ordres de grandeur de la complexité

Intérêt de la complexité

Bonnes pratiques

Rappels de complexité

Mesure de la complexité



HLIN608 – Algorithmique du texte.

Alban MANCHERON

En temps

- nombre d'opérations effectuées lors du déroulement de l'algorithme

En espace

- mémoire nécessaire pour le stockage des données calculées lors du déroulement de l'algorithme

Dans tous les cas :

- mesures en fonction de la forme des données en entrée du problème
- étude des cas particuliers
- ordre de grandeur (*e.g.*, logarithmique, linéaire, polynomial, exponentiel).

Rappels théoriques

Rappels de complexité

Mesure de la complexité

Ordres de grandeur de la complexité

Intérêt de la complexité

Bonnes pratiques

Rappels de complexité

Mesure de la complexité



HLIN608 – Algorithmique du texte.

Alban MANCHERON

En temps

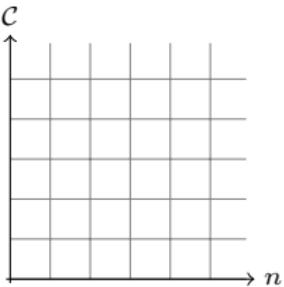
- nombre d'opérations effectuées lors du déroulement de l'algorithme

En espace

- mémoire nécessaire pour le stockage des données calculées lors du déroulement de l'algorithme

Dans tous les cas :

- mesures en fonction de la forme des données en entrée du problème
- étude des cas particuliers
- ordre de grandeur (*e.g.*, logarithmique, linéaire, polynomial, exponentiel).



Rappels théoriques

Rappels de complexité

Mesure de la complexité

Ordres de grandeur de la complexité

Intérêt de la complexité

Bonnes pratiques

Rappels de complexité

Mesure de la complexité



HLIN608 – Algorithmique du texte.

Alban MANCHERON

En temps

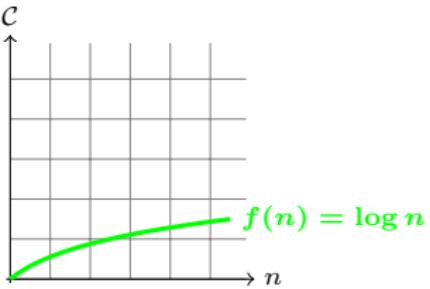
- nombre d'opérations effectuées lors du déroulement de l'algorithme

En espace

- mémoire nécessaire pour le stockage des données calculées lors du déroulement de l'algorithme

Dans tous les cas :

- mesures en fonction de la forme des données en entrée du problème
- étude des cas particuliers
- ordre de grandeur (*e.g.*, logarithmique, linéaire, polynomial, exponentiel).



Rappels théoriques

Rappels de complexité

Mesure de la complexité

Ordres de grandeur de la complexité

Intérêt de la complexité

Bonnes pratiques

Rappels de complexité

Mesure de la complexité



En temps

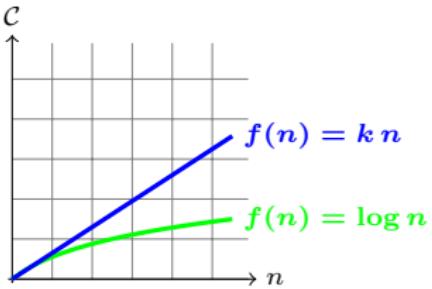
- nombre d'opérations effectuées lors du déroulement de l'algorithme

En espace

- mémoire nécessaire pour le stockage des données calculées lors du déroulement de l'algorithme

Dans tous les cas :

- mesures en fonction de la forme des données en entrée du problème
- étude des cas particuliers
- ordre de grandeur (*e.g.*, logarithmique, **linéaire**, polynomial, exponentiel).



Rappels théoriques

Rappels de complexité

Mesure de la complexité

Ordres de grandeur de la complexité

Intérêt de la complexité

Bonnes pratiques

Rappels de complexité

Mesure de la complexité



En temps

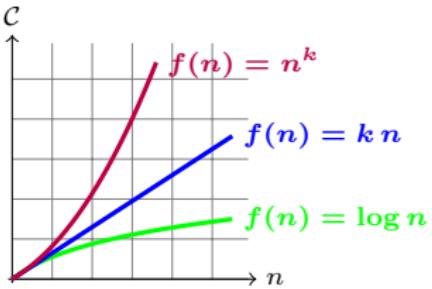
- nombre d'opérations effectuées lors du déroulement de l'algorithme

En espace

- mémoire nécessaire pour le stockage des données calculées lors du déroulement de l'algorithme

Dans tous les cas :

- mesures en fonction de la forme des données en entrée du problème
- étude des cas particuliers
- ordre de grandeur (*e.g.*, logarithmique, linéaire, **polynomial**, exponentiel).



Rappels théoriques

Rappels de complexité

Mesure de la complexité

Ordres de grandeur de la complexité

Intérêt de la complexité

Bonnes pratiques

Rappels de complexité

Mesure de la complexité



En temps

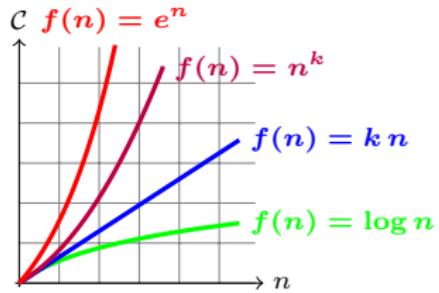
- nombre d'opérations effectuées lors du déroulement de l'algorithme

En espace

- mémoire nécessaire pour le stockage des données calculées lors du déroulement de l'algorithme

Dans tous les cas :

- mesures en fonction de la forme des données en entrée du problème
- étude des cas particuliers
- ordre de grandeur (*e.g.*, logarithmique, linéaire, polynomial, exponentiel).



Rappels théoriques

Rappels de complexité

Mesure de la complexité

Ordres de grandeur de la complexité

Intérêt de la complexité

Bonnes pratiques

Rappels de complexité

Mesure de la complexité



En temps

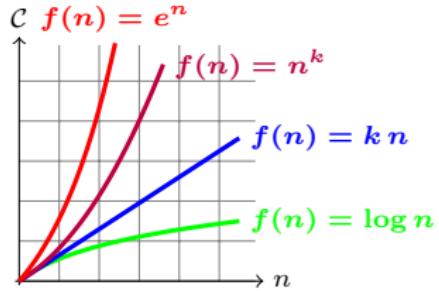
- nombre d'opérations effectuées lors du déroulement de l'algorithme

En espace

- mémoire nécessaire pour le stockage des données calculées lors du déroulement de l'algorithme

Dans tous les cas :

- mesures en fonction de la forme des données en entrée du problème
- étude des cas particuliers
- ordre de grandeur (*e.g.*, logarithmique, linéaire, polynomial, exponentiel).



Rappels théoriques

Rappels de complexité

Mesure de la complexité

Ordres de grandeur de la complexité

Intérêt de la complexité

Bonnes pratiques

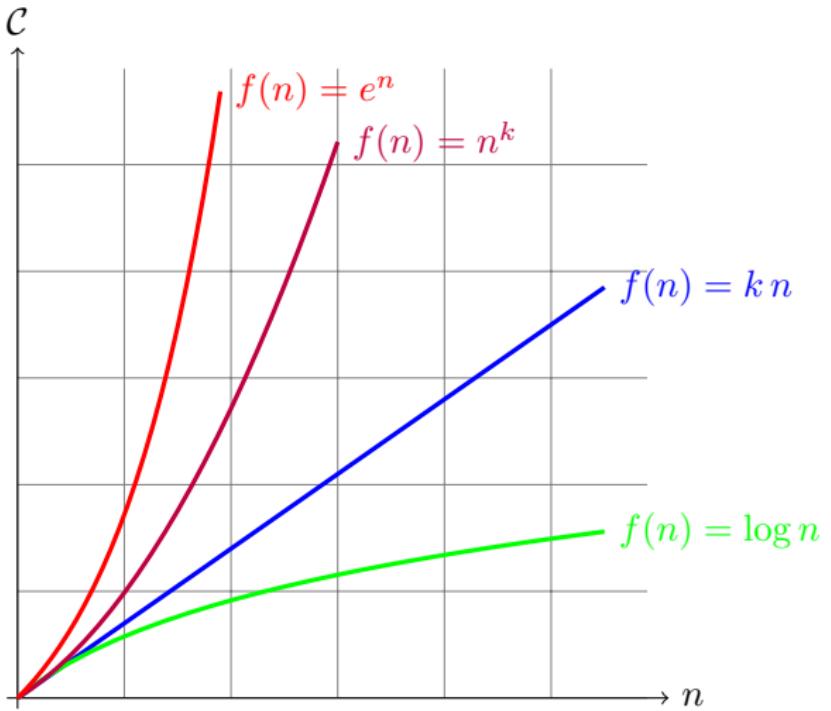
Rappels de complexité

Ordres de grandeur de la complexité



HLIN608 – Algorithmique du texte.

Alban MANCHERON



Rappels théoriques

Rappels de complexité

Mesure de la complexité

Ordres de grandeur de la complexité

Intérêt de la complexité

Bonnes pratiques

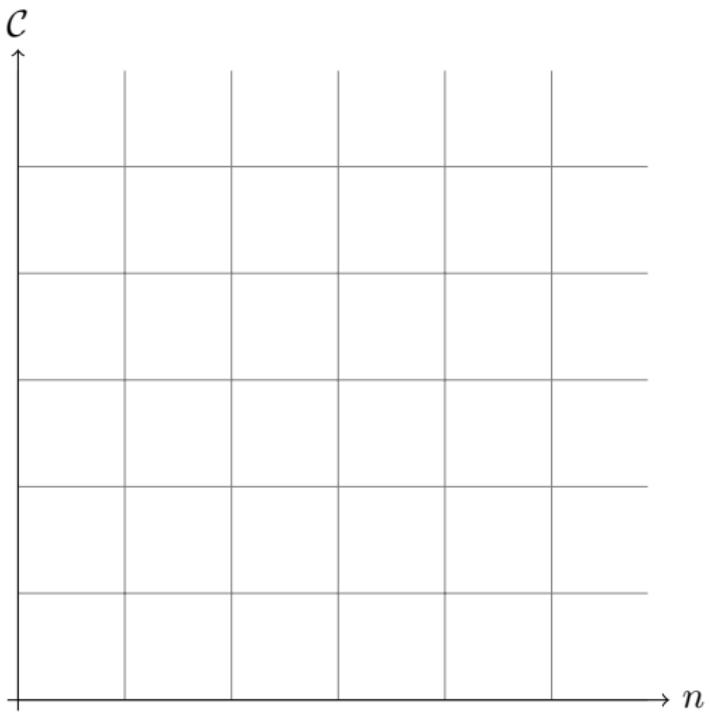
Rappels de complexité

Intérêt de la complexité



HLIN608 – Algorithmique du texte.

Alban MANCHERON



Rappels théoriques

Rappels de complexité

Mesure de la complexité

Ordres de grandeur de la complexité

Intérêt de la complexité

Bonnes pratiques

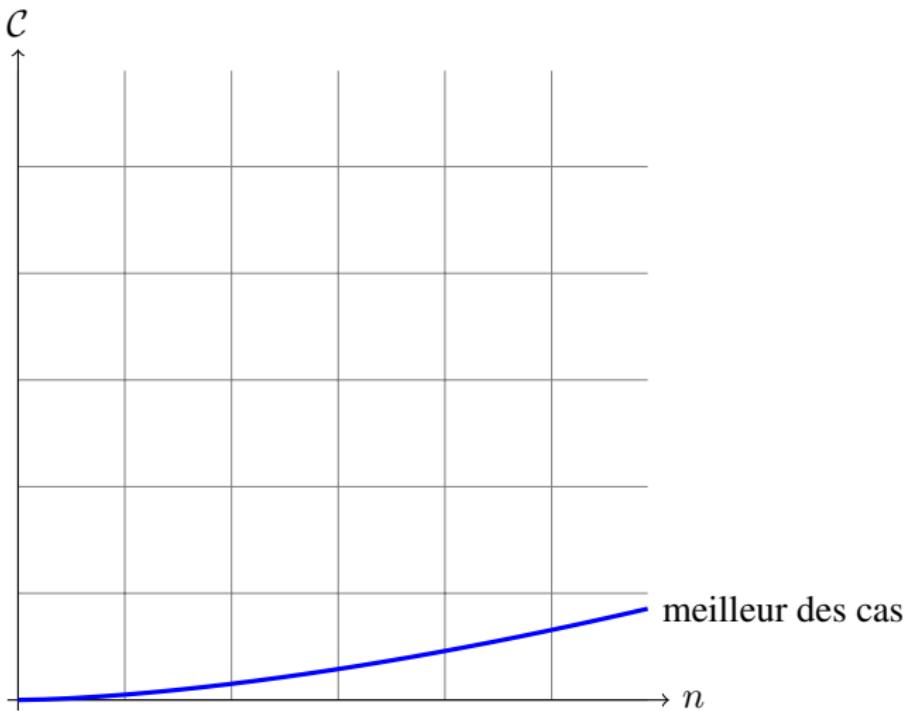
Rappels de complexité

Intérêt de la complexité



HLIN608 – Algorithmique du texte.

Alban MANCHERON



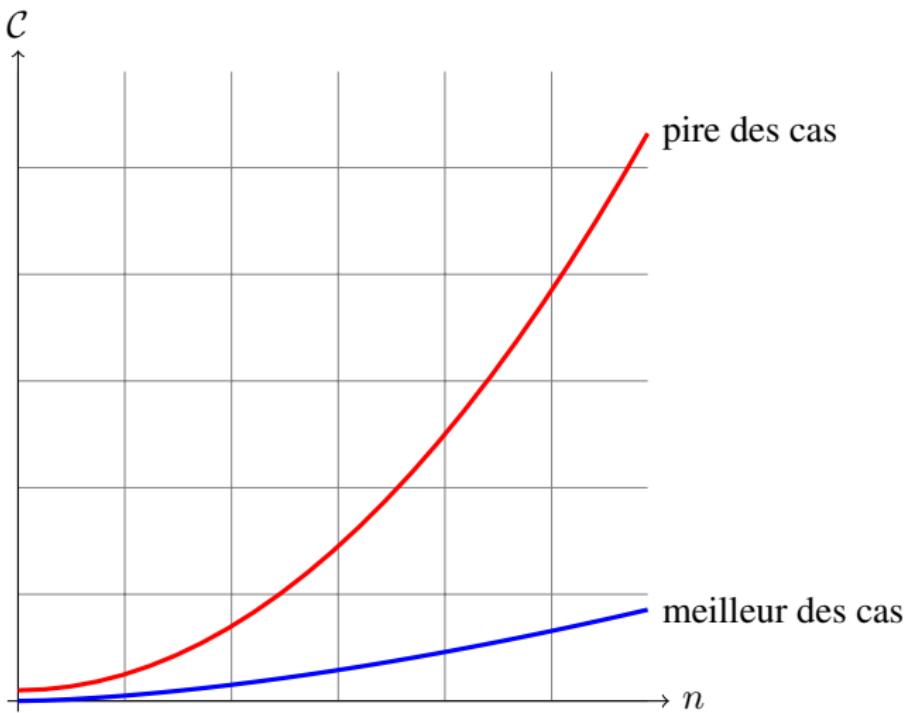
Rappels de complexité

Intérêt de la complexité



HLIN608 – Algorithmique du texte.

Alban MANCHERON



Rappels théoriques

Rappels de complexité

Mesure de la complexité

Ordres de grandeur de la complexité

Intérêt de la complexité

Bonnes pratiques

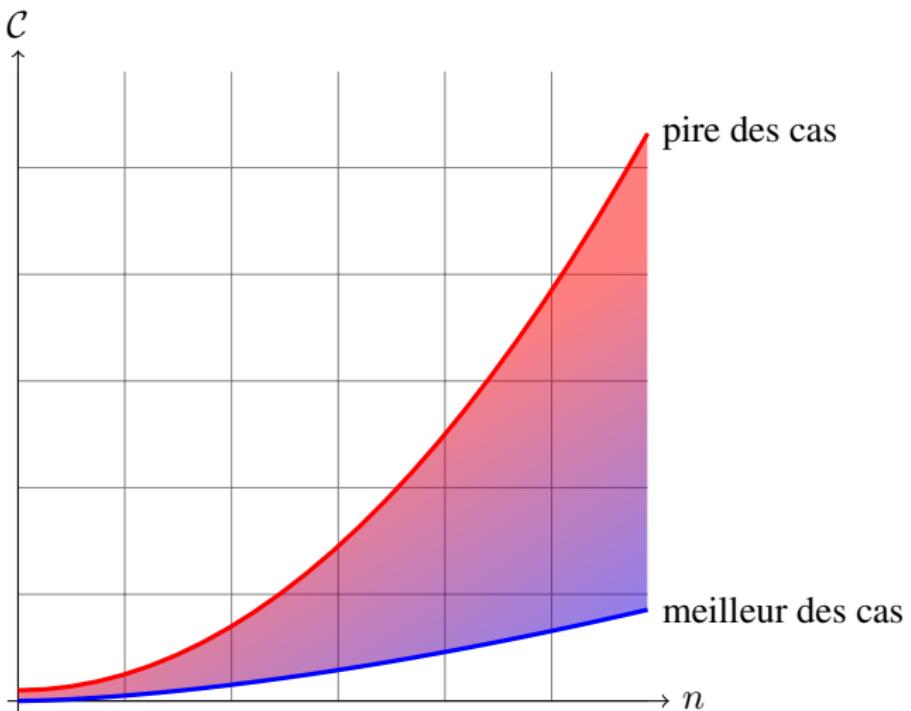
Rappels de complexité

Intérêt de la complexité



HLIN608 – Algorithmique du texte.

Alban MANCHERON



Rappels théoriques

Rappels de complexité

Mesure de la complexité

Ordres de grandeur de la complexité

Intérêt de la complexité

Bonnes pratiques

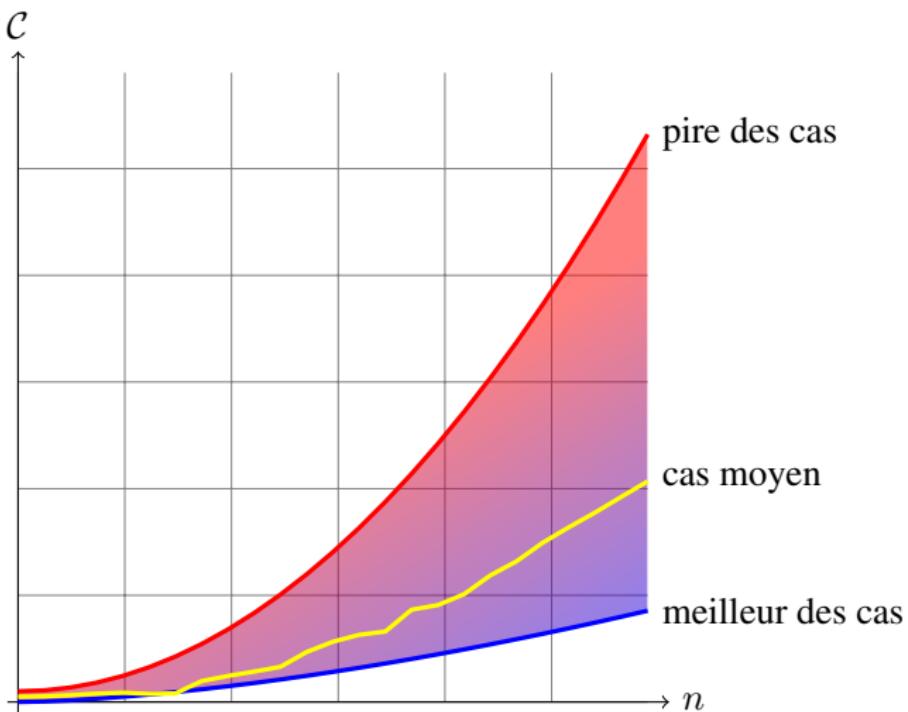
Rappels de complexité

Intérêt de la complexité



HLIN608 – Algorithmique du texte.

Alban MANCHERON



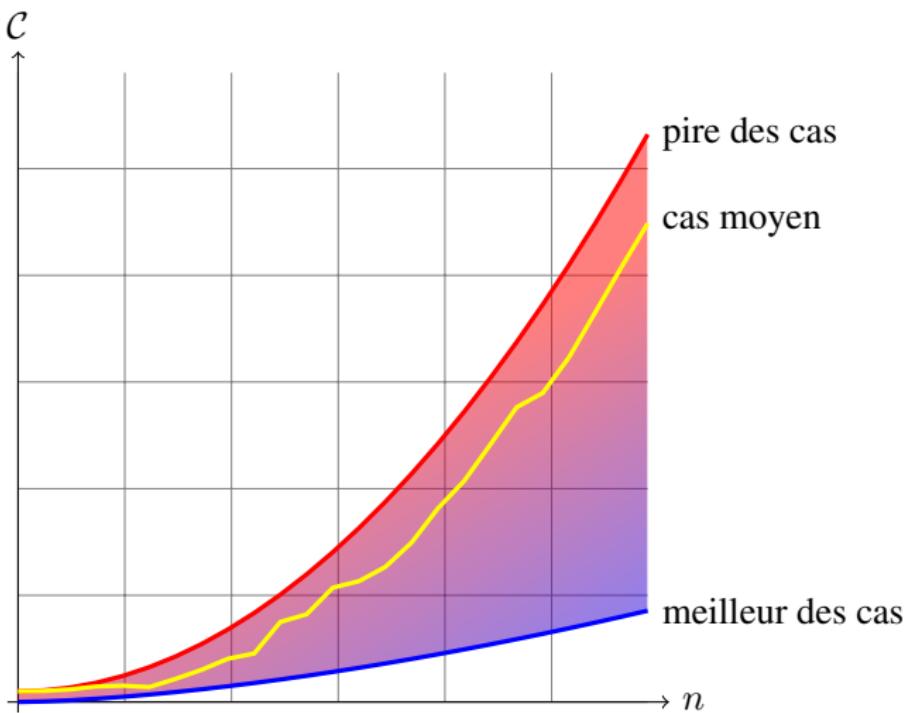
Rappels de complexité

Intérêt de la complexité



HLIN608 – Algorithmique du texte.

Alban MANCHERON



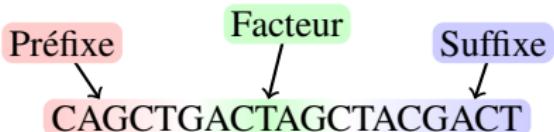
Rappels de complexité

Intérêt de la complexité



Vocabulaire

Alphabet $\longrightarrow \Sigma = \{A, C, G, T\}$



Notations

- Le mot vide est noté ε . Il est de longueur 0
- La longueur du mot $s = \alpha_1 \alpha_2 \cdots \alpha_n$ est égale à $|s| = n$
- Le mot $s[i..j]$ désigne le facteur commençant en i et terminant en j dans s
- On note $s[i]$ le symbole à la position i et $s[i..i] = s[i]$
- Le préfixe de longueur ℓ de s est $s[1..\ell]$
- Le suffixe de longueur ℓ de s est $s[n - \ell + 1..n]$

Questions

- Combien de préfixes/suffixes/facteurs pour un mot de longueur n ?
- Combien de sous-séquences pour un mot de longueur n (sachant qu'une sous-séquence s'obtient en enlevant entre 0 et n symboles de la séquence initiale) ?

HLIN608 – Algorithmique du texte.

Alban MANCHERON

Rappels théoriques

Rappels de complexité

Mesure de la complexité

Ordres de grandeur de la complexité

Intérêt de la complexité

Bonnes pratiques

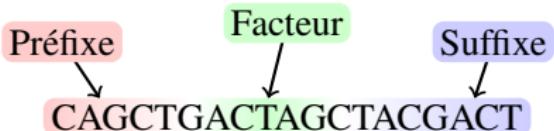
Rappels de complexité

Intérêt de la complexité



Vocabulaire

Alphabet $\longrightarrow \Sigma = \{A, C, G, T\}$



Notations

- Le mot vide est noté ε . Il est de longueur 0
- La longueur du mot $s = \alpha_1 \alpha_2 \cdots \alpha_n$ est égale à $|s| = n$
- Le mot $s[i..j]$ désigne le facteur commençant en i et terminant en j dans s
- On note $s[i]$ le symbole à la position i et $s[i..i] = s[i]$
- Le préfixe de longueur ℓ de s est $s[1..\ell]$
- Le suffixe de longueur ℓ de s est $s[n - \ell + 1..n]$

Questions

- Combien de préfixes/suffixes/facteurs pour un mot de longueur n ?
- Combien de sous-séquences pour un mot de longueur n (sachant qu'une sous-séquence s'obtient en enlevant entre 0 et n symboles de la séquence initiale) ?

HLIN608 – Algorithmique du texte.

Alban MANCHERON

Rappels théoriques

Rappels de complexité

Mesure de la complexité

Ordres de grandeur de la complexité

Intérêt de la complexité

Bonnes pratiques

Bonnes pratiques

Sommaire



HLIN608 – Algorithmique du texte.

Alban MANCHERON

Rappels théoriques

Rappels de complexité

Bonnes pratiques

II.1/ Rappels théoriques

II.2/ Rappels de complexité

II.3/ Bonnes pratiques



Bonnes pratiques

Réflexion

HLIN608 – Algorithmique du texte.

Alban MANCHERON

Rappels théoriques

Rappels de complexité

Bonnes pratiques



Bonnes pratiques

Réflexion

- Prendre le temps de l'analyse

HLIN608 – Algorithmique du texte.

Alban MANCHERON

Rappels théoriques

Rappels de complexité

Bonnes pratiques



Bonnes pratiques

Réflexion

- Prendre le temps de l'analyse
- Réfléchir aux structures de données

HLIN608 – Algorithmique du texte.

Alban MANCHERON

Rappels théoriques

Rappels de complexité

Bonnes pratiques



Bonnes pratiques

Réflexion

- Prendre le temps de l'analyse
- Réfléchir aux structures de données
- Estimer les paramètres de complexité du problème

HLIN608 – Algorithmique du texte.

Alban MANCHERON

Rappels théoriques

Rappels de complexité

Bonnes pratiques



Bonnes pratiques

Réflexion

- Prendre le temps de l'analyse
- Réfléchir aux structures de données
- Estimer les paramètres de complexité du problème

Programmation

HLIN608 – Algorithmique du texte.

Alban MANCHERON

Rappels théoriques

Rappels de complexité

Bonnes pratiques



Bonnes pratiques

Réflexion

- Prendre le temps de l'analyse
- Réfléchir aux structures de données
- Estimer les paramètres de complexité du problème

Programmation

- Coder proprement

HLIN608 – Algorithmique du texte.

Alban MANCHERON

Rappels théoriques

Rappels de complexité

Bonnes pratiques

Bonnes pratiques

Réflexion

- Prendre le temps de l'analyse
- Réfléchir aux structures de données
- Estimer les paramètres de complexité du problème

Programmation

- Coder proprement
- Choisir un langage approprié

HLIN608 – Algorithmique du texte.

Alban MANCHERON

Rappels théoriques

Rappels de complexité

Bonnes pratiques

Bonnes pratiques

Réflexion

- Prendre le temps de l'analyse
- Réfléchir aux structures de données
- Estimer les paramètres de complexité du problème

Programmation

- Coder proprement
- Choisir un langage approprié
- Réaliser des tests unitaires

HLIN608 – Algorithmique du texte.

Alban MANCHERON

Rappels théoriques

Rappels de complexité

Bonnes pratiques

Bonnes pratiques

Réflexion

- Prendre le temps de l'analyse
- Réfléchir aux structures de données
- Estimer les paramètres de complexité du problème

Programmation

- Coder proprement
- Choisir un langage approprié
- Réaliser des tests unitaires
- Appliquer le paradigme « Modèle-Vue-Contrôleur » (MVC)

HLIN608 – Algorithmique du texte.

Alban MANCHERON

Rappels théoriques

Rappels de complexité

Bonnes pratiques



Bonnes pratiques

Réflexion

- Prendre le temps de l'analyse
- Réfléchir aux structures de données
- Estimer les paramètres de complexité du problème

Programmation

- Coder proprement
- Choisir un langage approprié
- Réaliser des tests unitaires
- Appliquer le paradigme « Modèle-Vue-Contrôleur » (MVC)

Études

HLIN608 – Algorithmique du texte.

Alban MANCHERON

Rappels théoriques

Rappels de complexité

Bonnes pratiques

Bonnes pratiques

Réflexion

- Prendre le temps de l'analyse
- Réfléchir aux structures de données
- Estimer les paramètres de complexité du problème

HLIN608 – Algorithmique du texte.

Alban MANCHERON

Rappels théoriques

Rappels de complexité

Bonnes pratiques

Programmation

- Coder proprement
- Choisir un langage approprié
- Réaliser des tests unitaires
- Appliquer le paradigme « Modèle-Vue-Contrôleur » (MVC)

Études

- Poser des questions à bon escient

Bonnes pratiques

Réflexion

- Prendre le temps de l'analyse
- Réfléchir aux structures de données
- Estimer les paramètres de complexité du problème

HLIN608 – Algorithmique du texte.

Alban MANCHERON

Rappels théoriques

Rappels de complexité

Bonnes pratiques

Programmation

- Coder proprement
- Choisir un langage approprié
- Réaliser des tests unitaires
- Appliquer le paradigme « Modèle-Vue-Contrôleur » (MVC)

Études

- Poser des questions à bon escient
- Être curieux

Bonnes pratiques

Réflexion

- Prendre le temps de l'analyse
- Réfléchir aux structures de données
- Estimer les paramètres de complexité du problème

Programmation

- Coder proprement
- Choisir un langage approprié
- Réaliser des tests unitaires
- Appliquer le paradigme « Modèle-Vue-Contrôleur » (MVC)

Études

- Poser des questions à bon escient
- Être curieux
- Lire les documentations

Bonnes pratiques

Réflexion

- Prendre le temps de l'analyse
- Réfléchir aux structures de données
- Estimer les paramètres de complexité du problème

Programmation

- Coder proprement
- Choisir un langage approprié
- Réaliser des tests unitaires
- Appliquer le paradigme « Modèle-Vue-Contrôleur » (MVC)

Études

- Poser des questions à bon escient
- Être curieux
- Lire les documentations
- Regarder ce que les autres font

III – *Dot-plots et extraction de motifs* (v. 1.9)

Sommaire



HLIN608 – Algorithmique du texte.

Alban MANCHERON

Alignement visuel de séquences

Recherche de motif

Extraction de motif

III.1/ Alignement visuel de séquences

III.2/ Recherche de motif

III.3/ Extraction de motif

Alignment visuel de séquences

Sommaire



HLIN608 – Algorithmique du texte.

Alban MANCHERON

III.1/ Alignment visuel de séquences

- Présentation
- Intérêt de la comparaison de séquences
- Dot plot

III.2/ Recherche de motif

III.3/ Extraction de motif

Alignment visuel de séquences

Présentation

Intérêt de la comparaison de séquences

Dot plot

Recherche de motif

Extraction de motif

Alignment visuel de séquences

Présentation



Alignment de séquences

HLIN608 – Algorithmique du texte.

Alban MANCHERON

Alignment visuel de séquences

Présentation

Intérêt de la comparaison de séquences

Dot plot

Recherche de motif

Extraction de motif

Alignment visuel de séquences

Présentation



Alignment de séquences

- Problème représentatif de la bioinformatique

HLIN608 – Algorithmique du texte.

Alban MANCHERON

Alignment visuel de séquences

Présentation

Intérêt de la comparaison de séquences

Dot plot

Recherche de motif

Extraction de motif

Alignment visuel de séquences

Présentation



Alignment de séquences

- Problème représentatif de la bioinformatique
 - ⇒ Un des outils de bioinformatique les plus utilisés

HLIN608 – Algorithmique du texte.

Alban MANCHERON

Alignment visuel de séquences

Présentation

Intérêt de la comparaison de séquences

Dot plot

Recherche de motif

Extraction de motif

Alignment visuel de séquences

Présentation



Alignment de séquences

- Problème représentatif de la bioinformatique
 - ⇒ Un des outils de bioinformatique les plus utilisés
- Solution graphique et algorithmique

HLIN608 – Algorithmique du texte.

Alban MANCHERON

Alignment visuel de séquences

Présentation

Intérêt de la comparaison de séquences

Dot plot

Recherche de motif

Extraction de motif

Alignment visuel de séquences

Présentation



Alignment de séquences

- Problème représentatif de la bioinformatique
 - ⇒ Un des outils de bioinformatique les plus utilisés
- Solution graphique et algorithmique
- Algorithme de base élégant

HLIN608 – Algorithmique du texte.

Alban MANCHERON

Alignment visuel de séquences

Présentation

Intérêt de la comparaison de séquences

Dot plot

Recherche de motif

Extraction de motif

Alignment visuel de séquences

Présentation



Alignment de séquences

- Problème représentatif de la bioinformatique
 - ⇒ Un des outils de bioinformatique les plus utilisés
- Solution graphique et algorithmique
- Algorithme de base élégant
- Force et limite de l'informatique en biologie

HLIN608 – Algorithmique du texte.

Alban MANCHERON

Alignment visuel de séquences

Présentation

Intérêt de la comparaison de séquences

Dot plot

Recherche de motif

Extraction de motif

Alignment visuel de séquences

Présentation



Alignment de séquences

- Problème représentatif de la bioinformatique
 - ⇒ Un des outils de bioinformatique les plus utilisés
- Solution graphique et algorithmique
- Algorithme de base élégant
- Force et limite de l'informatique en biologie

Problématique

HLIN608 – Algorithmique du texte.

Alban MANCHERON

Alignment visuel de séquences

Présentation

Intérêt de la comparaison de séquences

Dot plot

Recherche de motif

Extraction de motif

Alignment visuel de séquences

Présentation



Alignment de séquences

- Problème représentatif de la bioinformatique
 - ⇒ Un des outils de bioinformatique les plus utilisés
- Solution graphique et algorithmique
- Algorithme de base élégant
- Force et limite de l'informatique en biologie

Problématique

- Recherche des similitudes entre plusieurs séquences

HLIN608 – Algorithmique du texte.

Alban MANCHERON

Alignment visuel de séquences

Présentation

Intérêt de la comparaison de séquences

Dot plot

Recherche de motif

Extraction de motif

Alignment visuel de séquences

Présentation



Alignment de séquences

- Problème représentatif de la bioinformatique
 - ⇒ Un des outils de bioinformatique les plus utilisés
- Solution graphique et algorithmique
- Algorithme de base élégant
- Force et limite de l'informatique en biologie

Problématique

- Recherche des similitudes entre plusieurs séquences
- Substitution, Insertion/Suppression (*InDel*)

HLIN608 – Algorithmique du texte.

Alban MANCHERON

Alignment visuel de séquences

Présentation

Intérêt de la comparaison de séquences

Dot plot

Recherche de motif

Extraction de motif

Alignment visuel de séquences

Présentation



Alignment de séquences

- Problème représentatif de la bioinformatique
 - ⇒ Un des outils de bioinformatique les plus utilisés
- Solution graphique et algorithmique
- Algorithme de base élégant
- Force et limite de l'informatique en biologie

Problématique

- Recherche des similitudes entre plusieurs séquences
- Substitution, Insertion/Suppression (*InDel*)

Exemple (POMME vs. GOMME)

P O M M E
G O M M E

HLIN608 – Algorithmique du texte.

Alban MANCHERON

Alignment visuel de séquences

Présentation

Intérêt de la comparaison de séquences

Dot plot

Recherche de motif

Extraction de motif

Alignment visuel de séquences

Présentation



Alignment de séquences

- Problème représentatif de la bioinformatique
 - ⇒ Un des outils de bioinformatique les plus utilisés
- Solution graphique et algorithmique
- Algorithme de base élégant
- Force et limite de l'informatique en biologie

Problématique

- Recherche des similitudes entre plusieurs séquences
- Substitution, Insertion/Suppression (*InDel*)

Exemple (POMME vs. GOMME)

P O M M E
G O M M E
* O M M E

HLIN608 – Algorithmique du texte.

Alban MANCHERON

Alignment visuel de séquences

Présentation

Intérêt de la comparaison de séquences

Dot plot

Recherche de motif

Extraction de motif

Alignment visuel de séquences

Présentation



HLIN608 – Algorithmique du texte.

Alban MANCHERON

Alignment de séquences

- Problème représentatif de la bioinformatique
 - ⇒ Un des outils de bioinformatique les plus utilisés
- Solution graphique et algorithmique
- Algorithme de base élégant
- Force et limite de l'informatique en biologie

Problématique

- Recherche des similitudes entre plusieurs séquences
- Substitution, Insertion/Suppression (*InDel*)

Exemple (OCTOBRE vs. OTARIE)

O C T O B R E
O T A R I E

Alignment visuel de séquences

Présentation

Intérêt de la comparaison de séquences

Dot plot

Recherche de motif

Extraction de motif

Alignment visuel de séquences

Présentation



HLIN608 – Algorithmique du texte.

Alban MANCHERON

Alignment de séquences

- Problème représentatif de la bioinformatique
 - ⇒ Un des outils de bioinformatique les plus utilisés
- Solution graphique et algorithmique
- Algorithme de base élégant
- Force et limite de l'informatique en biologie

Problématique

- Recherche des similitudes entre plusieurs séquences
- Substitution, Insertion/Suppression (*InDel*)

Exemple (OCTOBRE vs. OTARIE)

O	C	T	O	B	R	E
O	T	A	R	I	E	
O	*	*	*	*	*	-

Alignment visuel de séquences

Présentation

Intérêt de la comparaison de séquences

Dot plot

Recherche de motif

Extraction de motif

Alignment visuel de séquences

Présentation



Alignment de séquences

- Problème représentatif de la bioinformatique
 - ⇒ Un des outils de bioinformatique les plus utilisés
- Solution graphique et algorithmique
- Algorithme de base élégant
- Force et limite de l'informatique en biologie

Problématique

- Recherche des similitudes entre plusieurs séquences
- Substitution, Insertion/Suppression (*InDel*)

Exemple (OCTOBRE vs. OTARIE)

O	C	T	O	B	R	E
O	-	T	A	R	I	E
O	-	T	*	*	*	E

HLIN608 – Algorithmique du texte.

Alban MANCHERON

Alignment visuel de séquences

Présentation

Intérêt de la comparaison de séquences

Dot plot

Recherche de motif

Extraction de motif

Alignment visuel de séquences

Présentation



HLIN608 – Algorithmique du texte.

Alban MANCHERON

Alignment de séquences

- Problème représentatif de la bioinformatique
 - ⇒ Un des outils de bioinformatique les plus utilisés
- Solution graphique et algorithmique
- Algorithme de base élégant
- Force et limite de l'informatique en biologie

Problématique

- Recherche des similitudes entre plusieurs séquences
- Substitution, Insertion/Suppression (*InDel*)

Exemple (OCTOBRE vs. OTARIE)

O	C	T	O	B	R	E	
O	-	T	A	-	R	I	E
O	-	T	*	-	R	*	-

Alignment visuel de séquences

Présentation

Intérêt de la comparaison de séquences

Dot plot

Recherche de motif

Extraction de motif



Alignment de séquences

- Problème représentatif de la bioinformatique
 - ⇒ Un des outils de bioinformatique les plus utilisés
- Solution graphique et algorithmique
- Algorithme de base élégant
- Force et limite de l'informatique en biologie

Problématique

- Recherche des similitudes entre plusieurs séquences
- Substitution, Insertion/Suppression (*InDel*)

Exemple (OCTOBRE vs. OTARIE)

O	C	T	O	B	R	-	E
O	-	T	A	-	R	I	E
O	-	T	*	-	R	-	E

Alignment visuel de séquences

Présentation

Intérêt de la comparaison de séquences

Dot plot

Recherche de motif

Extraction de motif

Alignment visuel de séquences

Intérêt de la comparaison de séquences



Biologie moléculaire

HLIN608 – Algorithmique du texte.

Alban MANCHERON

Alignment visuel de séquences

Présentation

Intérêt de la comparaison de séquences

Dot plot

Recherche de motif

Extraction de motif

Alignment visuel de séquences

Intérêt de la comparaison de séquences



Biologie moléculaire

- Séquences (ADN, ARN, acides aminés, ...)

HLIN608 – Algorithmique du texte.

Alban MANCHERON

Alignment visuel de séquences

Présentation

Intérêt de la comparaison de séquences

Dot plot

Recherche de motif

Extraction de motif



Alignment visuel de séquences

Intérêt de la comparaison de séquences

Biologie moléculaire

- Séquences (ADN, ARN, acides aminés, ...)
- Recherche approchée dans une base de données

HLIN608 – Algorithmique du texte.

Alban MANCHERON

Alignment visuel de séquences

Présentation

Intérêt de la comparaison de séquences

Dot plot

Recherche de motif

Extraction de motif

Alignment visuel de séquences

Intérêt de la comparaison de séquences



Biologie moléculaire

- Séquences (ADN, ARN, acides aminés, ...)
- Recherche approchée dans une base de données
- Prédiction de structure

HLIN608 – Algorithmique du texte.

Alban MANCHERON

Alignment visuel de séquences

Présentation

Intérêt de la comparaison de séquences

Dot plot

Recherche de motif

Extraction de motif

Alignment visuel de séquences

Intérêt de la comparaison de séquences



Biologie moléculaire

- Séquences (ADN, ARN, acides aminés, ...)
- Recherche approchée dans une base de données
- Prédiction de structure
- Prédiction de fonction

HLIN608 – Algorithmique du texte.

Alban MANCHERON

Alignment visuel de séquences

Présentation

Intérêt de la comparaison de séquences

Dot plot

Recherche de motif

Extraction de motif

Alignment visuel de séquences

Intérêt de la comparaison de séquences



Biologie moléculaire

- Séquences (ADN, ARN, acides aminés, ...)
- Recherche approchée dans une base de données
- Prédition de structure
- Prédition de fonction
- Phylogénie (histoire évolutive des espèces)

HLIN608 – Algorithmique du texte.

Alban MANCHERON

Alignment visuel de séquences

Présentation

Intérêt de la comparaison de séquences

Dot plot

Recherche de motif

Extraction de motif

Alignment visuel de séquences

Intérêt de la comparaison de séquences



Biologie moléculaire

- Séquences (ADN, ARN, acides aminés, ...)
- Recherche approchée dans une base de données
- Prédition de structure
- Prédition de fonction
- Phylogénie (histoire évolutive des espèces)

Informatique

HLIN608 – Algorithmique du texte.

Alban MANCHERON

Alignment visuel de séquences

Présentation

Intérêt de la comparaison de séquences

Dot plot

Recherche de motif

Extraction de motif

Alignment visuel de séquences

Intérêt de la comparaison de séquences



Biologie moléculaire

- Séquences (ADN, ARN, acides aminés, ...)
- Recherche approchée dans une base de données
- Prédition de structure
- Prédition de fonction
- Phylogénie (histoire évolutive des espèces)

Informatique

- Comparaison de fichiers (1 ligne ~ 1 caractère)

HLIN608 – Algorithmique du texte.

Alban MANCHERON

Alignment visuel de séquences

Présentation

Intérêt de la comparaison de séquences

Dot plot

Recherche de motif

Extraction de motif

Alignment visuel de séquences

Intérêt de la comparaison de séquences



Biologie moléculaire

- Séquences (ADN, ARN, acides aminés, ...)
- Recherche approchée dans une base de données
- Prédition de structure
- Prédition de fonction
- Phylogénie (histoire évolutive des espèces)

Informatique

- Comparaison de fichiers (1 ligne \sim 1 caractère)
⇒ diff sous UNIX/Linux

HLIN608 – Algorithmique du texte.

Alban MANCHERON

Alignment visuel de séquences

Présentation

Intérêt de la comparaison de séquences

Dot plot

Recherche de motif

Extraction de motif

Alignment visuel de séquences

Intérêt de la comparaison de séquences



Biologie moléculaire

- Séquences (ADN, ARN, acides aminés, ...)
- Recherche approchée dans une base de données
- Prédition de structure
- Prédition de fonction
- Phylogénie (histoire évolutive des espèces)

Informatique

- Comparaison de fichiers (1 ligne \sim 1 caractère)
⇒ diff sous UNIX/Linux
- Recherche approchée dans un dictionnaire (correcteur orthographique, fautes de frappes), ...

HLIN608 – Algorithmique du texte.

Alban MANCHERON

Alignment visuel de séquences

Présentation

Intérêt de la comparaison de séquences

Dot plot

Recherche de motif

Extraction de motif

Alignment visuel de séquences

Intérêt de la comparaison de séquences



Biologie moléculaire

- Séquences (ADN, ARN, acides aminés, ...)
- Recherche approchée dans une base de données
- Prédition de structure
- Prédition de fonction
- Phylogénie (histoire évolutive des espèces)

Informatique

- Comparaison de fichiers (1 ligne \sim 1 caractère)
⇒ diff sous UNIX/Linux
- Recherche approchée dans un dictionnaire (correcteur orthographique, fautes de frappes), ...

Autres applications

HLIN608 – Algorithmique du texte.

Alban MANCHERON

Alignment visuel de séquences

Présentation

Intérêt de la comparaison de séquences

Dot plot

Recherche de motif

Extraction de motif



Alignment visuel de séquences

Intérêt de la comparaison de séquences

Biologie moléculaire

- Séquences (ADN, ARN, acides aminés, ...)
- Recherche approchée dans une base de données
- Prédition de structure
- Prédition de fonction
- Phylogénie (histoire évolutive des espèces)

Informatique

- Comparaison de fichiers (1 ligne \sim 1 caractère)
⇒ diff sous UNIX/Linux
- Recherche approchée dans un dictionnaire (correcteur orthographique, fautes de frappes), ...

Autres applications

- Reconnaissance vocale

HLIN608 – Algorithmique du texte.

Alban MANCHERON

Alignment visuel de séquences

Présentation

Intérêt de la comparaison de séquences

Dot plot

Recherche de motif

Extraction de motif

Alignment visuel de séquences

Intérêt de la comparaison de séquences



Biologie moléculaire

- Séquences (ADN, ARN, acides aminés, ...)
- Recherche approchée dans une base de données
- Prédition de structure
- Prédition de fonction
- Phylogénie (histoire évolutive des espèces)

Informatique

- Comparaison de fichiers (1 ligne \sim 1 caractère)
 \Rightarrow diff sous UNIX/Linux
- Recherche approchée dans un dictionnaire (correcteur orthographique, fautes de frappes), ...

Autres applications

- Reconnaissance vocale
- ...

HLIN608 – Algorithmique du texte.

Alban MANCHERON

Alignment visuel de séquences

Présentation

Intérêt de la comparaison de séquences

Dot plot

Recherche de motif

Extraction de motif

Alignment visuel de séquences

Dot plot [Exemple]



● Réponse visuelle au problème

HLIN608 – Algorithmique du texte.

Alban MANCHERON

Alignment visuel de séquences

Présentation

Intérêt de la comparaison de séquences

Dot plot

Exemple

Programmation

Filtrage

Recherche de motif

Extraction de motif



Alignment visuel de séquences

Dot plot [Exemple]

- Réponse visuelle au problème
- Largement utilisée

HLIN608 – Algorithmique du texte.

Alban MANCHERON

Alignment visuel de séquences

Présentation

Intérêt de la comparaison de séquences

Dot plot

Exemple

Programmation

Filtrage

Recherche de motif

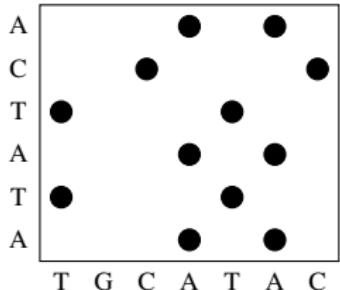
Extraction de motif

Alignment visuel de séquences

Dot plot [Exemple]



- Réponse visuelle au problème
- Largement utilisée



HLIN608 – Algorithmique du texte.

Alban MANCHERON

Alignment visuel de séquences

Présentation

Intérêt de la comparaison de séquences

Dot plot

Exemple

Programmation

Filtrage

Recherche de motif

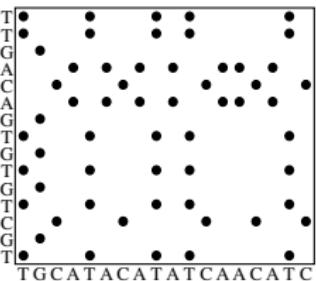
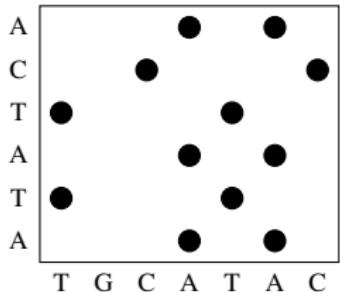
Extraction de motif

Alignement visuel de séquences

Dot plot [Exemple]



- Réponse visuelle au problème
- Largement utilisée



HLIN608 – Algorithmique du texte.

Alban MANCHERON

Alignement visuel de séquences

Présentation

Intérêt de la comparaison de séquences

Dot plot

Exemple

Programmation

Filtrage

Recherche de motif

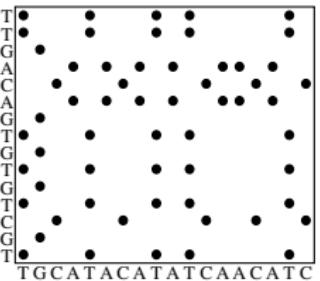
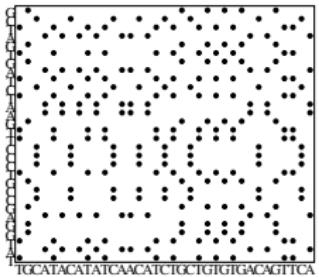
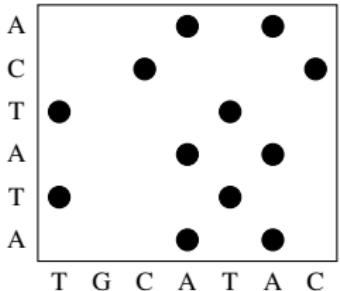
Extraction de motif

Alignement visuel de séquences

Dot plot [Exemple]



- Réponse visuelle au problème
 - Largement utilisée



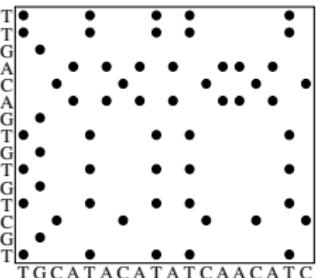
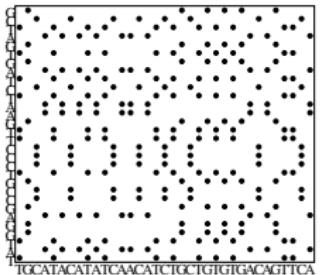
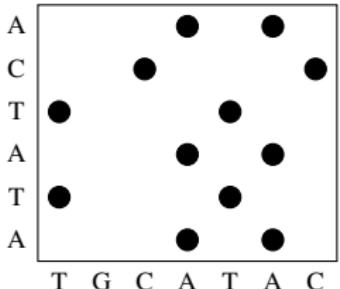
Extraction de motif

Alignement visuel de séquences

Dot plot [Exemple]



- Réponse visuelle au problème
- Largement utilisée



HLIN608 – Algorithmique du texte.

Alban MANCHERON

Alignement visuel de séquences

Présentation

Intérêt de la comparaison de séquences

Dot plot

Exemple

Programmation

Filtrage

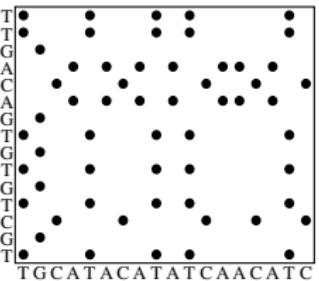
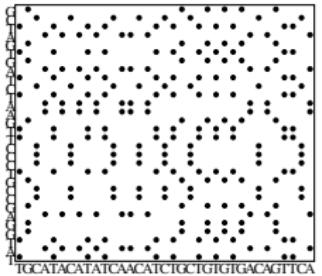
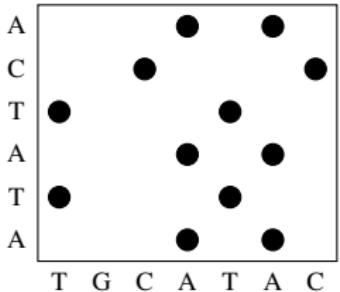
Recherche de motif

Extraction de motif

Alignement visuel de séquences

Dot plot [Exemple]

- Réponse visuelle au problème
- Largement utilisée



- Importance de la représentation graphique pour percevoir l'information
⇒ « Nettoyage » de l'affichage

HLIN608 – Algorithmique du texte.

Alban MANCHERON

Alignement visuel de séquences

Présentation

Intérêt de la comparaison de séquences

Dot plot

Exemple

Programmation

Filtrage

Recherche de motif

Extraction de motif

Alignement visuel de séquences

Dot plot [Programmation]



HLIN608 – Algorithmique du texte.

Alban MANCHERON

Alignement visuel de séquences

Présentation

Intérêt de la comparaison de séquences

Dot plot

Exemple

Programmation

Filtrage

Recherche de motif

Extraction de motif

DotPlot simple

```
1 int** dotPlot( const string &s1, const string &s2) {  
2     size_t 11 = s1.length(), 12 = s2.length();  
3     int** res = new int* [11];  
4     for (size_t i = 0; i < 11; i++) {  
5         res[i] = new int [12];  
6         for (size_t j = 0; j < 12; j++) {  
7             res[i][j] = (s1[i] == s2[j]);  
8         }  
9     }  
10    return res;  
11 }
```

Alignment visuel de séquences

Dot plot [Filtrage 1/3]



HLIN608 – Algorithmique du
texte.

Alban MANCHERON

- L'idée de base est de ne marquer que les « points » qui sont dans un motif potentiellement d'intérêt.

Alignment visuel de séquences

Présentation

Intérêt de la comparaison de séquences

Dot plot

Exemple

Programmation

Filtrage

Recherche de motif

Extraction de motif

Alignment visuel de séquences

Dot plot [Filtrage 1/3]



HLIN608 – Algorithmique du texte.

Alban MANCHERON

- L'idée de base est de ne marquer que les « points » qui sont dans un motif potentiellement d'intérêt.

⇒ $res[i][j] \leftarrow Vrai$ si et seulement si
 $s_1[i - k..i + k] = s_2[j - k..j + k]$ (pour $k > 0$)

Alignment visuel de séquences

Présentation

Intérêt de la comparaison de séquences

Dot plot

Exemple

Programmation

Filtrage

Recherche de motif

Extraction de motif

Alignment visuel de séquences

Dot plot [Filtrage 1/3]



HLIN608 – Algorithmique du texte.

Alban MANCHERON

- L'idée de base est de ne marquer que les « points » qui sont dans un motif potentiellement d'intérêt.
 - ⇒ $res[i][j] \leftarrow Vrai$ si et seulement si
 $s_1[i - k..i + k] = s_2[j - k..j + k]$ (pour $k > 0$)
 - ⇒ Filtrage moins brutal : seuil de similarité entre motifs

Alignment visuel de séquences

Présentation

Intérêt de la comparaison de séquences

Dot plot

Exemple

Programmation

Filtrage

Recherche de motif

Extraction de motif

Alignment visuel de séquences

Dot plot [Filtrage 1/3]



HLIN608 – Algorithmique du texte.

Alban MANCHERON

- L'idée de base est de ne marquer que les « points » qui sont dans un motif potentiellement d'intérêt.
 - ⇒ $res[i][j] \leftarrow Vrai$ si et seulement si
 $s_1[i - k..i + k] = s_2[j - k..j + k]$ (pour $k > 0$)
 - ⇒ Filtrage moins brutal : seuil de similarité entre motifs
- Attention aux limites des tableaux

Alignment visuel de séquences

Présentation

Intérêt de la comparaison de séquences

Dot plot

Exemple

Programmation

Filtrage

Recherche de motif

Extraction de motif

Alignment visuel de séquences

Dot plot [Filtrage 1/3]



HLIN608 – Algorithmique du texte.

Alban MANCHERON

- L'idée de base est de ne marquer que les « points » qui sont dans un motif potentiellement d'intérêt.
 - ⇒ $res[i][j] \leftarrow Vrai$ si et seulement si
 $s_1[i - k..i + k] = s_2[j - k..j + k]$ (pour $k > 0$)
 - ⇒ Filtrage moins brutal : seuil de similarité entre motifs
- Attention aux limites des tableaux
- Motifs complémentaires/inversés

Alignment visuel de séquences

Présentation

Intérêt de la comparaison de séquences

Dot plot

Exemple

Programmation

Filtrage

Recherche de motif

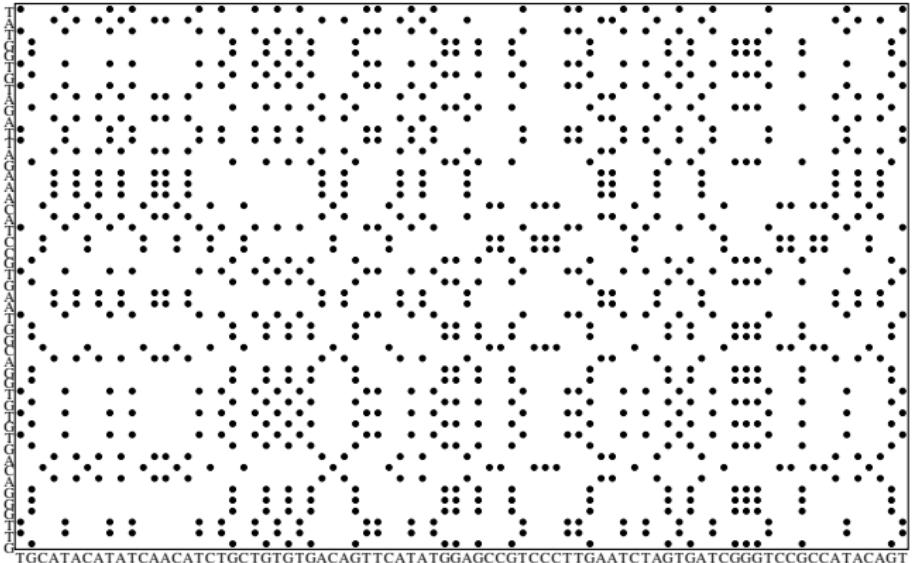
Extraction de motif

Alignement visuel de séquences

Dot plot [Filtrage 2/3]



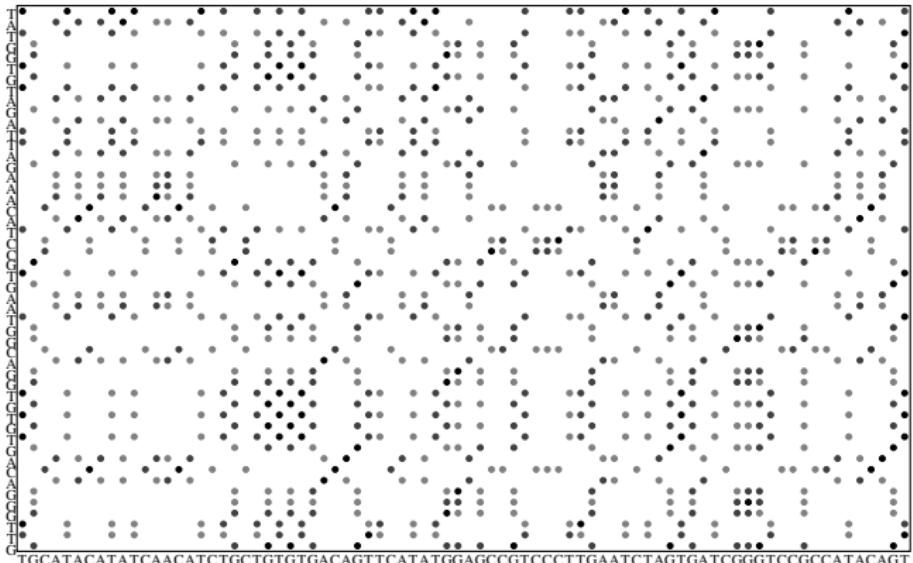
Dot plot (50 × 80) sans filtre



Alignement visuel de séquences

Dot plot [Filtrage 2/3]

Dot plot (50×80) avec filtrage [$k=1$]



HLIN608 – Algorithmique du texte.

Alban MANCHERON

Alignement visuel de séquences

Présentation

Intérêt de la comparaison de séquences

Dot plot

Exemple

Programmation

Filtrage

Recherche de motif

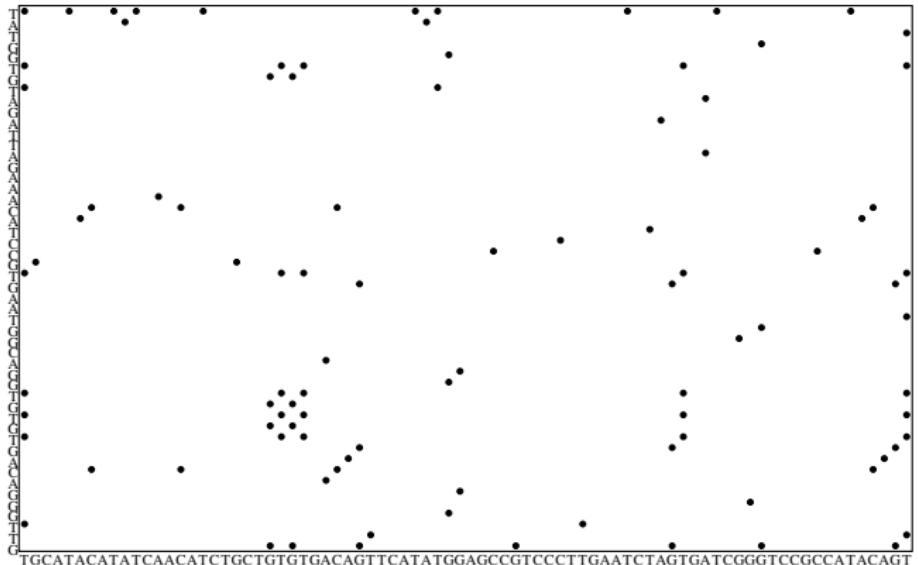
Extraction de motif

Alignement visuel de séquences

Dot plot [Filtrage 2/3]



Dot plot (50×80) avec filtrage [$k=1$]



HLIN608 – Algorithmique du texte.

Alban MANCHERON

Alignement visuel de séquences

Présentation

Intérêt de la comparaison de séquences

Dot plot

Exemple

Programmation

Filtrage

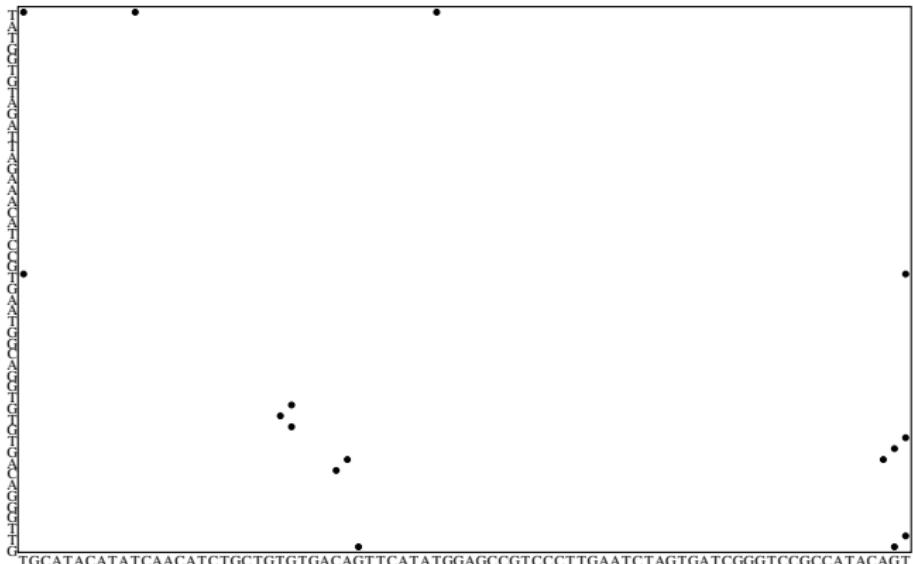
Recherche de motif

Extraction de motif

Alignement visuel de séquences

Dot plot [Filtrage 2/3]

Dot plot (50×80) avec filtrage [$k=2$]



HLIN608 – Algorithmique du texte.

Alban MANCHERON

Alignement visuel de séquences

Présentation

Intérêt de la comparaison de séquences

Dot plot

Exemple

Programmation

Filtrage

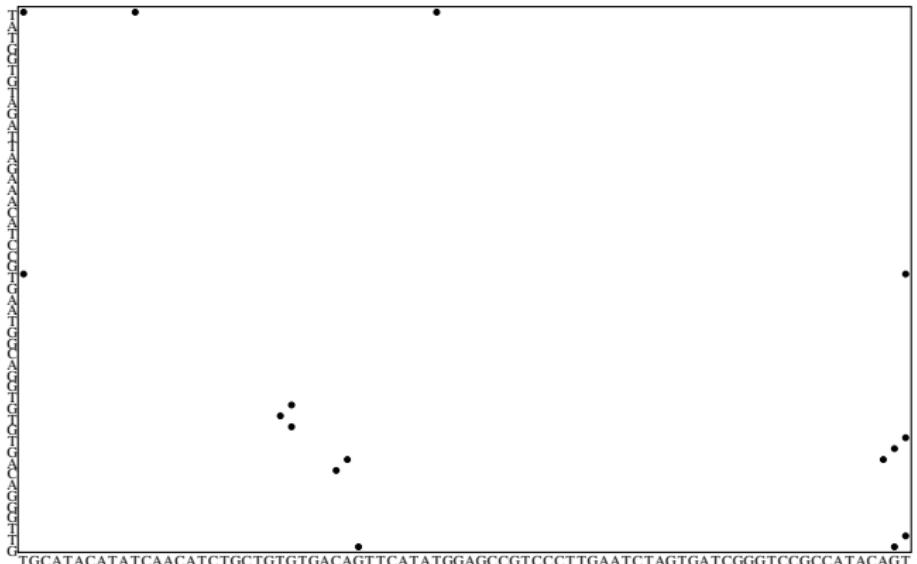
Recherche de motif

Extraction de motif

Alignement visuel de séquences

Dot plot [Filtrage 2/3]

Dot plot (50×80) avec filtrage [$k=2$]



- Complexité ?

HLIN608 – Algorithmique du texte.

Alban MANCHERON

Alignement visuel de séquences

Présentation

Intérêt de la comparaison de séquences

Dot plot

Exemple

Programmation

Filtrage

Recherche de motif

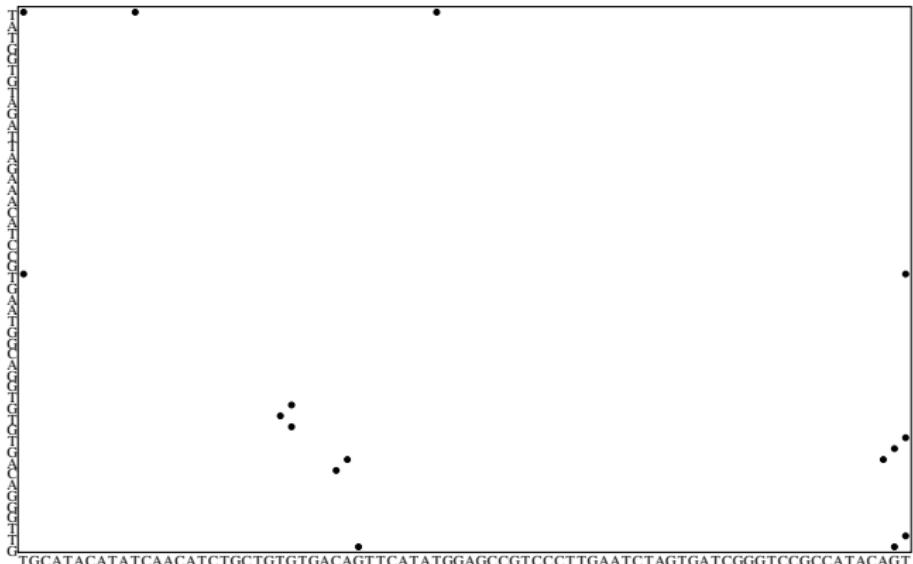
Extraction de motif

Alignement visuel de séquences

Dot plot [Filtrage 2/3]



Dot plot (50×80) avec filtrage [$k=2$]



Complexité ?

- ⇒ Dans le cas d'une identité totale on peut améliorer cette complexité
- ⇒ Le remplissage d'une ligne revient à faire une recherche de motif exact dans un mot

HLIN608 – Algorithmique du texte.

Alban MANCHERON

Alignement visuel de séquences

Présentation

Intérêt de la comparaison de séquences

Dot plot

Exemple

Programmation

Filtrage

Recherche de motif

Extraction de motif



Recherche de motif

Sommaire

III.1/ Alignement visuel de séquences

HLIN608 – Algorithmique du texte.

Alban MANCHERON

III.2/ Recherche de motif

- Algorithme naïf
- Illustration
- Algorithme KMP
- Calcul du bord maximal

III.3/ Extraction de motif

Alignement visuel de séquences

Recherche de motif

Algorithme naïf

Illustration

Algorithme KMP

Calcul du bord maximal

Extraction de motif

Recherche de motif

Algorithme naïf [Programmation]



HLIN608 – Algorithmique du texte.

Alban MANCHERON

Alignement visuel de séquences

Recherche de motif

Algorithme naïf

Programmation

Illustration

Algorithme KMP

Calcul du bord maximal

Extraction de motif

```
1 #include <iostream>
2 #include <string>
3 #include <libgen.h>
4
5 using namespace std;
6
7 int main(int argc, char** argv) {
8     string txt = "ACACTCCACT", mot = "ACT";
9     size_t l1 = txt.length(), l2 = mot.length();
10
11    for (size_t i=0; i < l1; i++) {
12        for (size_t j = 0; i + j < l1 && j < l2 && txt[i + j] == mot[j]; j++) {
13            if (j == l2 - 1) {
14                cout << "Found_at_location_" << i << endl;
15            }
16        }
17    }
18    return 0;
19 }
```

Recherche de motif

Algorithme naïf [Programmation]



Complexité ?

Recherche exacte

```
1 #include <iostream>
2 #include <string>
3 #include <libgen.h>
4
5 using namespace std;
6
7 int main(int argc, char** argv) {
8     string txt = "ACACTCCACT", mot = "ACT";
9     size_t l1 = txt.length(), l2 = mot.length();
10
11    for (size_t i=0; i < l1; i++) {
12        for (size_t j = 0; i + j < l1 && j < l2 && txt[i + j] == mot[j]; j++) {
13            if (j == l2 - 1) {
14                cout << "Found_at_location_" << i << endl;
15            }
16        }
17    }
18    return 0;
19 }
```

HLIN608 – Algorithmique du texte.

Alban MANCHERON

Alignement visuel de séquences

Recherche de motif

Algorithme naïf

Programmation

Illustration

Algorithme KMP

Calcul du bord maximal

Extraction de motif

Recherche de motif

Algorithme naïf [Programmation]



HLIN608 – Algorithmique du texte.

Alban MANCHERON

Alignement visuel de séquences

Recherche de motif

Algorithme naïf

Programmation

Illustration

Algorithme KMP

Calcul du bord maximal

Extraction de motif

Recher

Complexité ?

⇒ Pire de cas :

⇒ Meilleur de cas :

⇒ Cas moyen :

```
1 #include <iostream>
2 #include <string>
3 #include <libgen.h>
4
5 using namespace std;
6
7 int main(int argc, char** argv) {
8     string txt = "ACACTCCACT", mot = "ACT";
9     size_t ll = txt.length(), l2 = mot.length();
10
11    for (size_t i=0; i < ll; i++) {
12        for (size_t j = 0; i + j < ll && j < l2 && txt[i + j] == mot[j]; j++) {
13            if (j == l2 - 1) {
14                cout << "Found_at_location_" << i << endl;
15            }
16        }
17    }
18    return 0;
19 }
```

Recherche de motif

Algorithme naïf [Programmation]



HLIN608 – Algorithmique du texte.

Alban MANCHERON

Alignement visuel de séquences

Recherche de motif

Algorithme naïf

Programmation

Illustration

Algorithme KMP

Calcul du bord maximal

Extraction de motif

Recher

Complexité ?

⇒ Pire de cas : $O(nk)$

⇒ Meilleur de cas :

⇒ Cas moyen :

```
1 #include <iostream>
2 #include <string>
3 #include <libgen.h>
4
5 using namespace std;
6
7 int main(int argc, char** argv) {
8     string txt = "ACACTCCACT", mot = "ACT";
9     size_t l1 = txt.length(), l2 = mot.length();
10
11    for (size_t i=0; i < l1; i++) {
12        for (size_t j = 0; i + j < l1 && j < l2 && txt[i + j] == mot[j]; j++) {
13            if (j == l2 - 1) {
14                cout << "Found_at_location_" << i << endl;
15            }
16        }
17    }
18    return 0;
19 }
```

Recherche de motif

Algorithme naïf [Programmation]



HLIN608 – Algorithmique du texte.

Alban MANCHERON

Alignement visuel de séquences

Recherche de motif

Algorithme naïf

Programmation

Illustration

Algorithme KMP

Calcul du bord maximal

Extraction de motif

Recher

Complexité ?

⇒ Pire de cas : $O(nk)$

- Recherche de AAAAAAAC dans le mot
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA

⇒ Meilleur de cas :

⇒ Cas moyen :

```
1 #include <iostream>
2 #include <string>
3 #include <libgen.h>
4
5 using namespace std;
6
7 int main(int argc, char** argv) {
8     string txt = "ACACTCCACT", mot = "ACT";
9     size_t l1 = txt.length(), l2 = mot.length();
10
11    for (size_t i=0; i < l1; i++) {
12        for (size_t j = 0; i + j < l1 && j < l2 && txt[i + j] == mot[j]; j++) {
13            if (j == l2 - 1) {
14                cout << "Found_at_location_" << i << endl;
15            }
16        }
17    }
18    return 0;
19 }
```

Recherche de motif

Algorithme naïf [Programmation]



HLIN608 – Algorithmique du texte.

Alban MANCHERON

Alignement visuel de séquences

Recherche de motif

Algorithme naïf

Programmation

Illustration

Algorithme KMP

Calcul du bord maximal

Extraction de motif

Recher

Complexité ?

⇒ Pire de cas : $O(nk)$

- Recherche de AAAAAAAC dans le mot
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA

⇒ Meilleur de cas : $O(n)$

⇒ Cas moyen :

```
1 #include <iostream>
2 #include <string>
3 #include <libgen.h>
4
5 using namespace std;
6
7 int main(int argc, char** argv) {
8     string txt = "ACACTCCACT", mot = "ACT";
9     size_t l1 = txt.length(), l2 = mot.length();
10
11    for (size_t i=0; i < l1; i++) {
12        for (size_t j = 0; i + j < l1 && j < l2 && txt[i + j] == mot[j]; j++) {
13            if (j == l2 - 1) {
14                cout << "Found_at_location_" << i << endl;
15            }
16        }
17    }
18    return 0;
19 }
```

Recherche de motif

Algorithme naïf [Programmation]



HLIN608 – Algorithmique du texte.

Alban MANCHERON

Alignement visuel de séquences

Recherche de motif

Algorithme naïf

Programmation

Illustration

Algorithme KMP

Calcul du bord maximal

Extraction de motif

Recher

Complexité ?

⇒ Pire de cas : $O(nk)$

- Recherche de AAAAAAAC dans le mot
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA

⇒ Meilleur de cas : $O(n)$

- Recherche de CCCCCCCC dans le mot
AAAAAAAAAAAAAAAAAAAAAA

⇒ Cas moyen :

```
1 #include <iostream>
2 #include <string>
3 #include <libgen.h>
4
5 using namespace std;
6
7 int main(int argc, char** argv) {
8     string txt = "ACACTCCACT", mot = "ACT";
9     size_t l1 = txt.length(), l2 = mot.length();
10
11    for (size_t i=0; i < l1; i++) {
12        for (size_t j = 0; i + j < l1 && j < l2 && txt[i + j] == mot[j]; j++) {
13            if (j == l2 - 1) {
14                cout << "Found_at_location_" << i << endl;
15            }
16        }
17    }
18    return 0;
19 }
```

Recherche de motif

Algorithme naïf [Programmation]



HLIN608 – Algorithmique du texte.

Alban MANCHERON

Alignement visuel de séquences

Recherche de motif

Algorithme naïf

Programmation

Illustration

Algorithme KMP

Calcul du bord maximal

Extraction de motif

Recher

```
1 #include <iostream>
2 #include <string>
3 #include <libgen.h>
4
5 using namespace std;
6
7 int main(int argc, char** argv) {
8     string txt = "ACACTCCACT", mot = "ACT";
9     size_t ll = txt.length(), l2 = mot.length();
10
11    for (size_t i=0; i < ll; i++) {
12        for (size_t j = 0; i + j < ll && j < l2 && txt[i + j] == mot[j]; j++) {
13            if (j == l2 - 1) {
14                cout << "Found_at_location_" << i << endl;
15            }
16        }
17    }
18    return 0;
19 }
```

Complexité ?

⇒ Pire de cas : $O(nk)$

- Recherche de AAAAAAAC dans le mot
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA

⇒ Meilleur de cas : $O(n)$

- Recherche de CCCCCCCC dans le mot
AAAAAAAAAAAAAAAAAAAAAA

⇒ Cas moyen : dépend du nombre de match exacts

Recherche de motif

Illustration



HLIN608 – Algorithmique du
texte.

Alban MANCHERON

Recherche du mot COCO dans la séquence COCORICO

Alignement visuel de séquences

Recherche de motif

Algorithme naïf

Illustration

Algorithme KMP

Calcul du bord maximal

Extraction de motif

Recherche de motif

Illustration



HLIN608 – Algorithmique du texte.

Alban MANCHERON

Recherche du mot COCO dans la séquence COCORICO

C O C O R I C O

Alignement visuel de séquences

Recherche de motif

Algorithme naïf

Illustration

Algorithme KMP

Calcul du bord maximal

Extraction de motif



Recherche de motif

Illustration

HLIN608 – Algorithmique du texte.

Alban MANCHERON

Recherche du mot COCO dans la séquence COCORICO

C O C O R I C O

$i \leftarrow 0$

Alignement visuel de séquences

Recherche de motif

Algorithmme naïf

Illustration

Algorithmme KMP

Calcul du bord maximal

Extraction de motif



Recherche de motif

Illustration

HLIN608 – Algorithmique du texte.

Alban MANCHERON

Recherche du mot COCO dans la séquence COCORICO

C O C O R I C O

$i \leftarrow 0$ C

Alignement visuel de séquences

Recherche de motif

Algorithmme naïf

Illustration

Algorithmme KMP

Calcul du bord maximal

Extraction de motif



Recherche de motif

Illustration

HLIN608 – Algorithmique du texte.

Alban MANCHERON

Recherche du mot COCO dans la séquence COCORICO

C O C O R I C O

$i \leftarrow 0$ C O

Alignement visuel de séquences

Recherche de motif

Algorithm naïf

Illustration

Algorithm KMP

Calcul du bord maximal

Extraction de motif

Recherche de motif

Illustration



HLIN608 – Algorithmique du texte.

Alban MANCHERON

Recherche du mot COCO dans la séquence COCORICO

C O C O R I C O
 $i \leftarrow 0$ C O C

Alignement visuel de séquences

Recherche de motif

Algorithmme naïf

Illustration

Algorithmme KMP

Calcul du bord maximal

Extraction de motif

Recherche de motif

Illustration



HLIN608 – Algorithmique du texte.

Alban MANCHERON

Recherche du mot COCO dans la séquence COCORICO

C O C O R I C O
 $i \leftarrow 0$ C O C O

Alignement visuel de séquences

Recherche de motif

Algorithmme naïf

Illustration

Algorithmme KMP

Calcul du bord maximal

Extraction de motif



Recherche de motif

Illustration

HLIN608 – Algorithmique du texte.

Alban MANCHERON

Recherche du mot COCO dans la séquence COCORICO

C O C O R I C O

$i \leftarrow 0$ C O C O

$i \leftarrow 1$

Alignement visuel de séquences

Recherche de motif

Algorithmique naïf

Illustration

Algorithm KMP

Calcul du bord maximal

Extraction de motif



Recherche de motif

Illustration

HLIN608 – Algorithmique du texte.

Alban MANCHERON

Recherche du mot COCO dans la séquence COCORICO

C O C O R I C O

$i \leftarrow 0$ C O C O

$i \leftarrow 1$ C

Alignement visuel de séquences

Recherche de motif

Algorithme naïf

Illustration

Algorithme KMP

Calcul du bord maximal

Extraction de motif



Recherche de motif

Illustration

HLIN608 – Algorithmique du texte.

Alban MANCHERON

Recherche du mot COCO dans la séquence COCORICO

C O C O R I C O

$i \leftarrow 0$ C O C O

$i \leftarrow 1$ C

$i \leftarrow 2$

Alignement visuel de séquences

Recherche de motif

Algorithme naïf

Illustration

Algorithme KMP

Calcul du bord maximal

Extraction de motif



Recherche de motif

Illustration

HLIN608 – Algorithmique du texte.

Alban MANCHERON

Recherche du mot COCO dans la séquence COCORICO

C O C O R I C O

$i \leftarrow 0$ C O C O

$i \leftarrow 1$ C

$i \leftarrow 2$ C

Alignement visuel de séquences

Recherche de motif

Algorithme naïf

Illustration

Algorithme KMP

Calcul du bord maximal

Extraction de motif



Recherche de motif

Illustration

HLIN608 – Algorithmique du texte.

Alban MANCHERON

Recherche du mot COCO dans la séquence COCORICO

C O C O R I C O	
$i \leftarrow 0$	C O C O
$i \leftarrow 1$	C
$i \leftarrow 2$	C O

Alignement visuel de séquences

Recherche de motif

Algorithme naïf

Illustration

Algorithme KMP

Calcul du bord maximal

Extraction de motif



Recherche de motif

Illustration

HLIN608 – Algorithmique du texte.

Alban MANCHERON

Recherche du mot COCO dans la séquence COCORICO

C O C O R I C O
 $i \leftarrow 0$ C O C O
 $i \leftarrow 1$ C
 $i \leftarrow 2$ C O C

Alignement visuel de séquences

Recherche de motif

Algorithme naïf

Illustration

Algorithme KMP

Calcul du bord maximal

Extraction de motif

Recherche de motif

Illustration



HLIN608 – Algorithmique du texte.

Alban MANCHERON

Recherche du mot COCO dans la séquence COCORICO

C O C O R I C O	
$i \leftarrow 0$	C O C O
$i \leftarrow 1$	C
$i \leftarrow 2$	C O C
$i \leftarrow 3$	

Alignement visuel de séquences

Recherche de motif

Algorithme naïf

Illustration

Algorithme KMP

Calcul du bord maximal

Extraction de motif

Recherche de motif

Illustration



HLIN608 – Algorithmique du texte.

Alban MANCHERON

Recherche du mot COCO dans la séquence COCORICO

C O C O R I C O	
$i \leftarrow 0$	C O C O
$i \leftarrow 1$	C
$i \leftarrow 2$	C O C
$i \leftarrow 3$	C

Alignement visuel de séquences

Recherche de motif

Algorithme naïf

Illustration

Algorithme KMP

Calcul du bord maximal

Extraction de motif

Recherche de motif

Illustration



HLIN608 – Algorithmique du texte.

Alban MANCHERON

Recherche du mot COCO dans la séquence COCORICO

C O C O R I C O	
$i \leftarrow 0$	C O C O
$i \leftarrow 1$	C
$i \leftarrow 2$	C O C
$i \leftarrow 3$	C
$i \leftarrow 4$	

Alignement visuel de séquences

Recherche de motif

Algorithme naïf

Illustration

Algorithme KMP

Calcul du bord maximal

Extraction de motif



Recherche de motif

Illustration

HLIN608 – Algorithmique du texte.

Alban MANCHERON

Recherche du mot COCO dans la séquence COCORICO

C O C O R I C O	
$i \leftarrow 0$	C O C O
$i \leftarrow 1$	C
$i \leftarrow 2$	C O C
$i \leftarrow 3$	C
$i \leftarrow 4$	C

Alignement visuel de séquences

Recherche de motif

Algorithme naïf

Illustration

Algorithme KMP

Calcul du bord maximal

Extraction de motif



Recherche de motif

Illustration

HLIN608 – Algorithmique du texte.

Alban MANCHERON

Recherche du mot COCO dans la séquence COCORICO

C O C O R I C O	
$i \leftarrow 0$	C O C O
$i \leftarrow 1$	C
$i \leftarrow 2$	C O C
$i \leftarrow 3$	C
$i \leftarrow 4$	C
$i \leftarrow 5$	

Alignement visuel de séquences

Recherche de motif

Algorithme naïf

Illustration

Algorithme KMP

Calcul du bord maximal

Extraction de motif



Recherche de motif

Illustration

HLIN608 – Algorithmique du texte.

Alban MANCHERON

Recherche du mot COCO dans la séquence COCORICO

C O C O R I C O	
$i \leftarrow 0$	C O C O
$i \leftarrow 1$	C
$i \leftarrow 2$	C O C
$i \leftarrow 3$	C
$i \leftarrow 4$	C
$i \leftarrow 5$	C

Alignement visuel de séquences

Recherche de motif

Algorithme naïf

Illustration

Algorithme KMP

Calcul du bord maximal

Extraction de motif

Recherche de motif

Illustration



HLIN608 – Algorithmique du texte.

Alban MANCHERON

Recherche du mot COCO dans la séquence COCORICO

C O C O R I C O	
$i \leftarrow 0$	C O C O
$i \leftarrow 1$	C
$i \leftarrow 2$	C O C
$i \leftarrow 3$	C
$i \leftarrow 4$	C
$i \leftarrow 5$	C
$i \leftarrow 6$	

Alignement visuel de séquences

Recherche de motif

Algorithme naïf

Illustration

Algorithme KMP

Calcul du bord maximal

Extraction de motif

Recherche de motif

Illustration



HLIN608 – Algorithmique du texte.

Alban MANCHERON

Recherche du mot COCO dans la séquence COCORICO

C O C O R I C O	
$i \leftarrow 0$	C O C O
$i \leftarrow 1$	C
$i \leftarrow 2$	C O C
$i \leftarrow 3$	C
$i \leftarrow 4$	C
$i \leftarrow 5$	C
$i \leftarrow 6$	C

Alignement visuel de séquences

Recherche de motif

Algorithme naïf

Illustration

Algorithme KMP

Calcul du bord maximal

Extraction de motif

Recherche de motif

Illustration



HLIN608 – Algorithmique du texte.

Alban MANCHERON

Recherche du mot COCO dans la séquence COCORICO

C O C O R I C O	
$i \leftarrow 0$	C O C O
$i \leftarrow 1$	C
$i \leftarrow 2$	C O C
$i \leftarrow 3$	C
$i \leftarrow 4$	C
$i \leftarrow 5$	C
$i \leftarrow 6$	C O

Alignement visuel de séquences

Recherche de motif

Algorithme naïf

Illustration

Algorithme KMP

Calcul du bord maximal

Extraction de motif

Recherche de motif

Illustration



HLIN608 – Algorithmique du texte.

Alban MANCHERON

Recherche du mot COCO dans la séquence COCORICO

C O C O R I C O	
$i \leftarrow 0$	C O C O
$i \leftarrow 1$	C
$i \leftarrow 2$	C O C
$i \leftarrow 3$	C
$i \leftarrow 4$	C
$i \leftarrow 5$	C
$i \leftarrow 6$	C O
$i \leftarrow 7$	

Alignement visuel de séquences

Recherche de motif

Algorithme naïf

Illustration

Algorithme KMP

Calcul du bord maximal

Extraction de motif

Recherche de motif

Illustration



HLIN608 – Algorithmique du texte.

Alban MANCHERON

Recherche du mot COCO dans la séquence COCORICO

C O C O R I C O	
$i \leftarrow 0$	C O C O
$i \leftarrow 1$	C
$i \leftarrow 2$	C O C
$i \leftarrow 3$	C
$i \leftarrow 4$	C
$i \leftarrow 5$	C
$i \leftarrow 6$	C O
$i \leftarrow 7$	C

Alignement visuel de séquences

Recherche de motif

Algorithme naïf

Illustration

Algorithme KMP

Calcul du bord maximal

Extraction de motif



Recherche de motif

Algorithme KMP

Principe Éviter de refaire des comparaisons superflues

HLIN608 – Algorithmique du texte.

Alban MANCHERON

Alignement visuel de séquences

Recherche de motif

Algorithme naïf

Illustration

Algorithme KMP

Calcul du bord maximal

Extraction de motif

Recherche de motif

Algorithme KMP



Principe Éviter de refaire des comparaisons superflues

- ⇒ Sauter les itérations où aucune correspondance n'est possible
- ⇒ Une correspondance à l'itération suivante chevauchera partiellement la correspondance courante

HLIN608 – Algorithmique du texte.

Alban MANCHERON

Alignement visuel de séquences

Recherche de motif

Algorithmme naïf

Illustration

Algorithmme KMP

Calcul du bord maximal

Extraction de motif

Recherche de motif

Algorithme KMP



Principe Éviter de refaire des comparaisons superflues

- ⇒ Sauter les itérations où aucune correspondance n'est possible
- ⇒ Une correspondance à l'itération suivante chevauchera partiellement la correspondance courante

HLIN608 – Algorithmique du texte.

Alban MANCHERON

Alignement visuel de séquences

Recherche de motif

Algorithme naïf

Illustration

Algorithme KMP

Calcul du bord maximal

Extraction de motif

KNUTH-MORRIS-PRATT

Recherche de motif

Algorithme KMP



Principe Éviter de refaire des comparaisons superflues

- ⇒ Sauter les itérations où aucune correspondance n'est possible
- ⇒ Une correspondance à l'itération suivante chevauchera partiellement la correspondance courante

HLIN608 – Algorithmique du texte.

Alban MANCHERON

Alignement visuel de séquences

Recherche de motif

Algorithme naïf

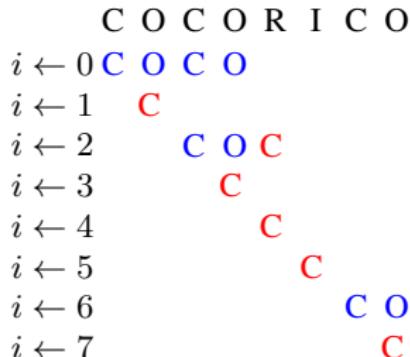
Illustration

Algorithme KMP

Calcul du bord maximal

Extraction de motif

KNUTH-MORRIS-PRATT



Recherche de motif

Algorithme KMP



Principe Éviter de refaire des comparaisons superflues

- ⇒ Sauter les itérations où aucune correspondance n'est possible
- ⇒ Une correspondance à l'itération suivante chevauchera partiellement la correspondance courante

HLIN608 – Algorithmique du texte.

Alban MANCHERON

Alignement visuel de séquences

Recherche de motif

Algorithme naïf

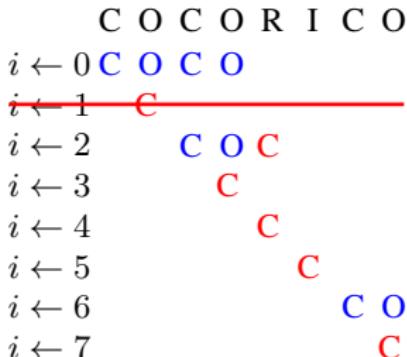
Illustration

Algorithme KMP

Calcul du bord maximal

Extraction de motif

KNUTH-MORRIS-PRATT



Recherche de motif

Algorithme KMP



Principe Éviter de refaire des comparaisons superflues

- ⇒ Sauter les itérations où aucune correspondance n'est possible
- ⇒ Une correspondance à l'itération suivante chevauchera partiellement la correspondance courante

HLIN608 – Algorithmique du texte.

Alban MANCHERON

Alignement visuel de séquences

Recherche de motif

Algorithme naïf

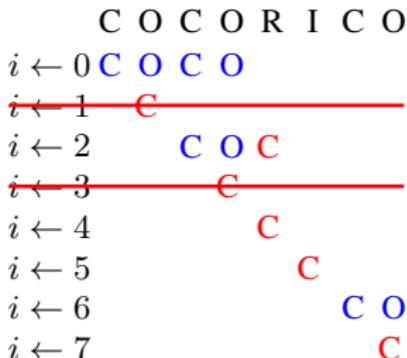
Illustration

Algorithme KMP

Calcul du bord maximal

Extraction de motif

KNUTH-MORRIS-PRATT



Recherche de motif

Algorithme KMP



Principe Éviter de refaire des comparaisons superflues

- ⇒ Sauter les itérations où aucune correspondance n'est possible
- ⇒ Une correspondance à l'itération suivante chevauchera partiellement la correspondance courante

HLIN608 – Algorithmique du texte.

Alban MANCHERON

Alignement visuel de séquences

Recherche de motif

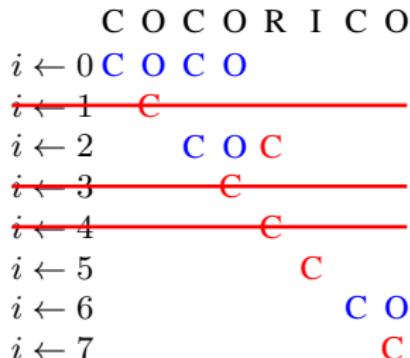
Algorithme naïf

Illustration

Algorithme KMP

Calcul du bord maximal

Extraction de motif



KNUTH-MORRIS-PRATT

Recherche de motif

Algorithme KMP



Principe Éviter de refaire des comparaisons superflues

- ⇒ Sauter les itérations où aucune correspondance n'est possible
- ⇒ Une correspondance à l'itération suivante chevauchera partiellement la correspondance courante

HLIN608 – Algorithmique du texte.

Alban MANCHERON

Alignement visuel de séquences

Recherche de motif

Algorithme naïf

Illustration

Algorithme KMP

Calcul du bord maximal

Extraction de motif

KNUTH-MORRIS-PRATT

C	O	C	O	R	I	C	O
$i \leftarrow 0$	C	O	C	O			
$i \leftarrow 1$	C						
$i \leftarrow 2$		C	O	C			
$i \leftarrow 3$		C					
$i \leftarrow 4$			C				
$i \leftarrow 5$				C			
$i \leftarrow 6$					C	O	
$i \leftarrow 7$						C	

Recherche de motif

Algorithme KMP



HLIN608 – Algorithmique du texte.

Alban MANCHERON

Bord maximal [overlap]

Le chevauchement de deux mots x et y est la taille du plus long suffixe propre de x qui est également préfixe de y .

Alignement visuel de séquences

Recherche de motif

Algorithme naïf

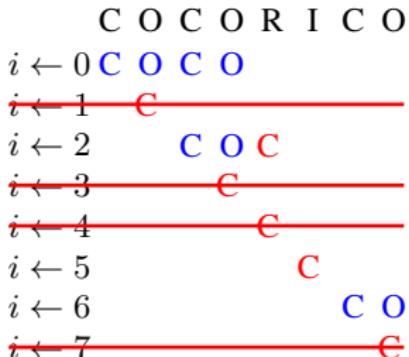
Illustration

Algorithme KMP

Calcul du bord maximal

Extraction de motif

KNUTH-MORRIS-PRATT



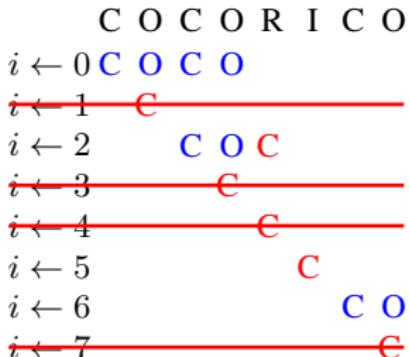


Bord maximal [overlap]

Le chevauchement de deux mots x et y est la taille du plus long suffixe propre de x qui est également préfixe de y .

- Si le préfixe $m[1..\ell]$ correspond à $s[i..i + \ell - 1]$ et que $m[\ell + 1] \neq s[i + \ell]$, où aura lieu (au mieux) la prochaine correspondance dans $s[i..|s|]$?

KNUTH-MORRIS-PRATT



Alignement visuel de séquences

Recherche de motif

Algorithme naïf

Illustration

Algorithme KMP

Calcul du bord maximal

Extraction de motif

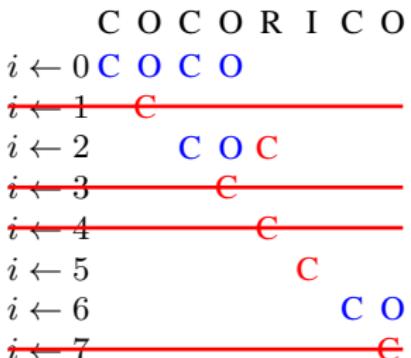
Recherche de motif

Algorithme KMP



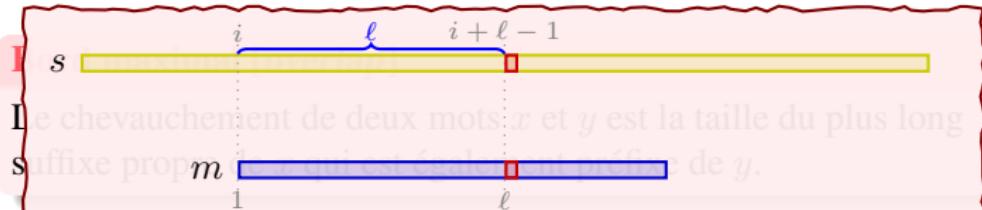
- Si le préfixe $m[1..\ell]$ correspond à $s[i..i + \ell - 1]$ et que $m[\ell + 1] \neq s[i + \ell]$, où aura lieu (au mieux) la prochaine correspondance dans $s[i..|s|]$?

KNUTH-MORRIS-PRATT



Recherche de motif

Algorithme KMP



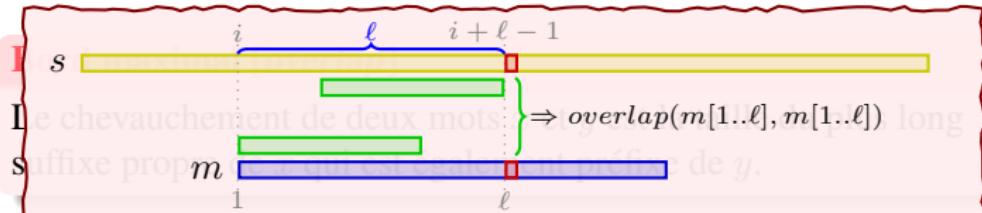
- Si le préfixe $m[1..\ell]$ correspond à $s[i..i + \ell - 1]$ et que $m[\ell + 1] \neq s[i + \ell]$, où aura lieu (au mieux) la prochaine correspondance dans $s[i..|s|]$?

KNUTH-MORRIS-PRATT

	C	O	C	O	R	I	C	O
$i \leftarrow 0$	C	O	C	O				
$i \leftarrow 1$	C							
$i \leftarrow 2$		C	O	C				
$i \leftarrow 3$			C					
$i \leftarrow 4$				C				
$i \leftarrow 5$					C			
$i \leftarrow 6$						C	O	
$i \leftarrow 7$							C	

Recherche de motif

Algorithme KMP



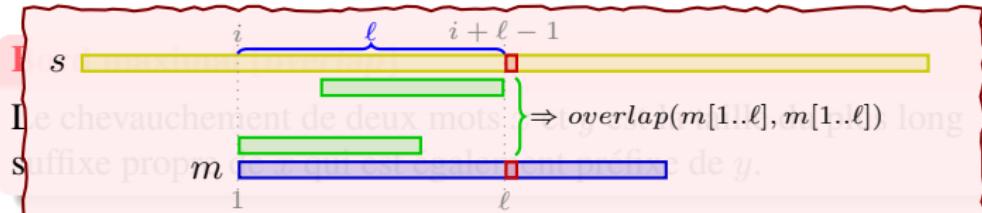
- Si le préfixe $m[1..\ell]$ correspond à $s[i..i + \ell - 1]$ et que $m[\ell + 1] \neq s[i + \ell]$, où aura lieu (au mieux) la prochaine correspondance dans $s[i..|s|]$?

KNUTH-MORRIS-PRATT

	C	O	C	O	R	I	C	O
$i \leftarrow 0$	C	O	C	O				
$i \leftarrow 1$	C							
$i \leftarrow 2$		C	O	C				
$i \leftarrow 3$		C						
$i \leftarrow 4$			C					
$i \leftarrow 5$				C				
$i \leftarrow 6$				C	O			
$i \leftarrow 7$					C			

Recherche de motif

Algorithme KMP



- Si le préfixe $m[1..\ell]$ correspond à $s[i..i + \ell - 1]$ et que $m[\ell + 1] \neq s[i + \ell]$, où aura lieu (au mieux) la prochaine correspondance dans $s[i..|s|]$?
 \Rightarrow à la position $i + \ell - overlap(m[1..\ell], m[1..\ell])$

C	O	C	O	R	I	C	O
$i \leftarrow 0$	C	O	C	O			
$i \leftarrow 1$	C						
$i \leftarrow 2$		C	O	C			
$i \leftarrow 3$		C					
$i \leftarrow 4$			C				
$i \leftarrow 5$				C			
$i \leftarrow 6$					C	O	
$i \leftarrow 7$						C	

KNUTH-MORRIS-PRATT

HLIN608 – Algorithmique du texte.

Alban MANCHERON

Alignement visuel de séquences

Recherche de motif

Algorithmique naïf

Illustration

Algorithm KMP

Calcul du bord maximal

Extraction de motif

Recherche de motif

Calcul du bord maximal



• Calcul dynamique

HLIN608 – Algorithmique du texte.

Alban MANCHERON

Alignement visuel de séquences

Recherche de motif

Algorithmme naïf

Illustration

Algorithmme KMP

Calcul du bord maximal

Extraction de motif



Recherche de motif

Calcul du bord maximal

• Calcul dynamique

Calcul dynamique du chevauchement

```
1 int Overlap(const string &seq) {
2     bool ok = false;
3     size_t lg, l = lg = seq.length();
4     while (!ok && l--) {
5         ok = true;
6         for (size_t i = 0; ok && (i < l); i++) {
7             ok = (seq[i] == seq[lg-l+i]);
8         }
9     }
10    return l;
11 }
```

HLIN608 – Algorithmique du texte.

Alban MANCHERON

Alignement visuel de séquences

Recherche de motif

Algorithme naïf

Illustration

Algorithme KMP

Calcul du bord maximal

Extraction de motif

Recherche de motif

Calcul du bord maximal



• Calcul dynamique

Calcul dynamique du chevauchement

```
1 int Overlap(const string &seq) {
2     bool ok = false;
3     size_t lg, l = lg = seq.length();
4     while (!ok && l--) {
5         ok = true;
6         for (size_t i = 0; ok && (i < l); i++) {
7             ok = (seq[i] == seq[lg-l+i]);
8         }
9     }
10    return l;
11 }
```



HLIN608 – Algorithmique du texte.

Alban MANCHERON

Alignement visuel de séquences

Recherche de motif

Algorithme naïf

Illustration

Algorithme KMP

Calcul du bord maximal

Extraction de motif

Recherche de motif

Calcul du bord maximal



• Calcul dynamique

Calcul dynamique du chevauchement

```
1 int Overlap(const string &seq) {
2     bool ok = false;
3     size_t lg, l = lg = seq.length();
4     while (!ok && l--) {
5         ok = true;
6         for (size_t i = 0; ok && (i < l); i++) {
7             ok = (seq[i] == seq[lg-l+i]);
8         }
9     }
10    return l;
11 }
```



HLIN608 – Algorithmique du texte.

Alban MANCHERON

Alignement visuel de séquences

Recherche de motif

Algorithme naïf

Illustration

Algorithme KMP

Calcul du bord maximal

Extraction de motif

Recherche de motif

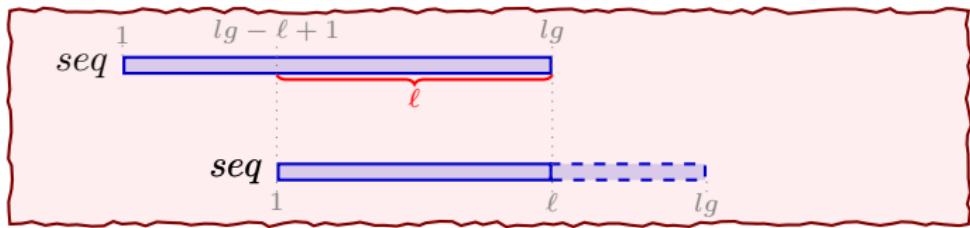
Calcul du bord maximal



• Calcul dynamique

Calcul dynamique du chevauchement

```
1 int Overlap(const string &seq) {
2     bool ok = false;
3     size_t lg, l = lg = seq.length();
4     while (!ok && l--) {
5         ok = true;
6         for (size_t i = 0; ok && (i < l); i++) {
7             ok = (seq[i] == seq[lg-l+i]);
8         }
9     }
10    return l;
11 }
```



Recherche de motif

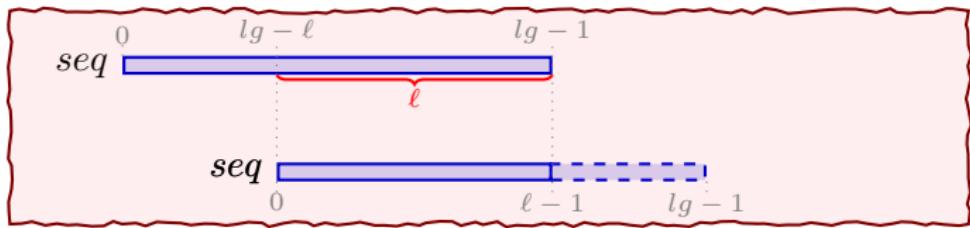
Calcul du bord maximal



• Calcul dynamique

Calcul dynamique du chevauchement

```
1 int Overlap(const string &seq) {
2     bool ok = false;
3     size_t lg, l = lg = seq.length();
4     while (!ok && l--) {
5         ok = true;
6         for (size_t i = 0; ok && (i < l); i++) {
7             ok = (seq[i] == seq[lg-1+i]);
8         }
9     }
10    return l;
11 }
```



HLIN608 – Algorithmique du texte.

Alban MANCHERON

Alignement visuel de séquences

Recherche de motif

Algorithme naïf

Illustration

Algorithme KMP

Calcul du bord maximal

Extraction de motif

Recherche de motif

Calcul du bord maximal

• Calcul dynamique

Calcul dynamique du chevauchement

```
1 int Overlap(const string &seq) {
2     bool ok = false;
3     size_t lg, l = lg = seq.length();
4     while (!ok && l--) {
5         ok = true;
6         for (size_t i = 0; ok && (i < l); i++) {
7             ok = (seq[i] == seq[lg-l+i]);
8         }
9     }
10    return l;
11 }
```

⇒ redondance !!!

⇒ Complexité temporelle de la recherche en $O(n m^2)$

Recherche de motif

Calcul du bord maximal



• Pré-traitement simple

Calcul des chevauchements de tous les préfixes

```
1 int* Overlap(const string &seq) {  
2     size_t lg = seq.length();  
3     size_t* overlap = new int[lg + 1];  
4     overlap[0] = (size_t) -1;  
5     for (size_t i = 1; i < lg; i++) {  
6         overlap[i] = Overlap(seq.substr(0, i));  
7     }  
8     return overlap;  
9 }
```

HLIN608 – Algorithmique du texte.

Alban MANCHERON

Alignement visuel de séquences

Recherche de motif

Algorithme naïf

Illustration

Algorithme KMP

Calcul du bord maximal

Extraction de motif

Recherche de motif

Calcul du bord maximal



• Pré-traitement simple

Calcul des chevauchements de tous les préfixes

```
1 int* Overlap(const string &seq) {  
2     size_t lg = seq.length();  
3     size_t* overlap = new int[lg + 1];  
4     overlap[0] = (size_t) -1;  
5     for (size_t i = 1; i < lg; i++) {  
6         overlap[i] = Overlap(seq.substr(0, i));  
7     }  
8     return overlap;  
9 }
```

⇒ Complexité temporelle en $O(n + m^3)$

HLIN608 – Algorithmique du texte.

Alban MANCHERON

Alignement visuel de séquences

Recherche de motif

Algorithme naïf

Illustration

Algorithme KMP

Calcul du bord maximal

Extraction de motif

Recherche de motif

Calcul du bord maximal

• Pré-traitement « malin »

Calcul efficace des chevauchements de tous les préfixes

```
1 int* Overlap( const string &seq) {
2     size_t lg = seq.length();
3     size_t* overlap = new int[lg + 1];
4     overlap[0] = (size_t)-1;
5     for (size_t i = 0; i < lg; i++) {
6         overlap[i + 1] = overlap[i] + 1;
7         while (overlap[i + 1] > 0 &&
8                 seq[i] != seq[overlap[i + 1] - 1]) {
9             overlap[i + 1] = overlap[overlap[i + 1] - 1] + 1;
10        }
11    }
12    return overlap;
13 }
```

Recherche de motif

Calcul du bord maximal

• Pré-traitement « malin »

Calcul efficace des chevauchements de tous les préfixes

```
1 int* Overlap( const string &seq) {
2     size_t lg = seq.length();
3     size_t* overlap = new int[lg + 1];
4     overlap[0] = (size_t)-1;
5     for (size_t i = 0; i < lg; i++) {
6         overlap[i + 1] = overlap[i] + 1;
7         while (overlap[i + 1] > 0 &&
8                 seq[i] != seq[overlap[i + 1] - 1]) {
9             overlap[i + 1] = overlap[overlap[i + 1] - 1] + 1;
10        }
11    }
12    return overlap;
13 }
```

⇒ Complexité temporelle en $O(n + m)$

Recherche de motif

Calcul du bord maximal

• Pré-traitement « malin »

Calcul efficace des chevauchements de tous les préfixes

```
1 int* Overlap( const string &seq) {
2     size_t lg = seq.length();
3     size_t* overlap = new int[lg + 1];
4     overlap[0] = (size_t)-1;
5     for (size_t i = 0; i < lg; i++) {
6         overlap[i + 1] = overlap[i] + 1;
7         while (overlap[i + 1] > 0 &&
8                 seq[i] != seq[overlap[i + 1] - 1]) {
9             overlap[i + 1] = overlap[overlap[i + 1] - 1] + 1;
10        }
11    }
12    return overlap;
13 }
```

	i	0	1	2	3	4
	seq	C	O	C	O	\0
$overlap$						

Recherche de motif

Calcul du bord maximal



● Pré-traitement « malin »

Calcul efficace des chevauchements de tous les préfixes

```
1 int* Overlap( const string &seq) {
2     size_t lg = seq.length();
3     size_t* overlap = new int[lg + 1];
4     overlap[0] = (size_t)-1;
5     for (size_t i = 0; i < lg; i++) {
6         overlap[i + 1] = overlap[i] + 1;
7         while (overlap[i + 1] > 0 &&
8                 seq[i] != seq[overlap[i + 1] - 1]) {
9             overlap[i + 1] = overlap[overlap[i + 1] - 1] + 1;
10        }
11    }
12    return overlap;
13 }
```

i	0	1	2	3	4	
seq	C	O	C	O	\0	ligne 4
$overlap$	-1					

Recherche de motif

Calcul du bord maximal



● Pré-traitement « malin »

Calcul efficace des chevauchements de tous les préfixes

```
1 int* Overlap( const string &seq) {
2     size_t lg = seq.length();
3     size_t* overlap = new int[lg + 1];
4     overlap[0] = (size_t)-1;
5     for (size_t i = 0; i < lg; i++) {
6         overlap[i + 1] = overlap[i] + 1;
7         while (overlap[i + 1] > 0 &&
8                 seq[i] != seq[overlap[i + 1] - 1]) {
9             overlap[i + 1] = overlap[overlap[i + 1] - 1] + 1;
10        }
11    }
12    return overlap;
13 }
```

Alignement visuel de séquences

Recherche de motif

Algorithm naïf

Illustration

Algorithm KMP

Calcul du bord maximal

Extraction de motif

i	0	1	2	3	4	
seq	C	O	C	O	\0	ligne 5
$overlap$	-1					

Recherche de motif

Calcul du bord maximal

- Pré-traitement « malin »

Calcul efficace des chevauchements de tous les préfixes

```

1 int* Overlap( const string &seq) {
2     size_t lg = seq.length();
3     size_t* overlap = new int[lg + 1];
4     overlap[0] = (size_t)-1;
5     for (size_t i = 0; i < lg; i++) {
6         overlap[i + 1] = overlap[i] + 1;
7         while (overlap[i + 1] > 0 &&
8                 seq[i] != seq[overlap[i + 1] - 1]) {
9             overlap[i + 1] = overlap[overlap[i + 1] - 1] + 1;
10        }
11    }
12    return overlap;
13 }
```

i	0	1	2	3	4	
seq	C	O	C	O	\0	ligne 6
$overlap$	-1	0				

Recherche de motif

Calcul du bord maximal

- Pré-traitement « malin »

Calcul efficace des chevauchements de tous les préfixes

```

1 int* Overlap( const string &seq) {
2     size_t lg = seq.length();
3     size_t* overlap = new int[lg + 1];
4     overlap[0] = (size_t)-1;
5     for (size_t i = 0; i < lg; i++) {
6         overlap[i + 1] = overlap[i] + 1;
7         while (overlap[i + 1] > 0 &&
8                 seq[i] != seq[overlap[i + 1] - 1]) {
9             overlap[i + 1] = overlap[overlap[i + 1] - 1] + 1;
10        }
11    }
12    return overlap;
13 }
```

<i>i</i>	0	1	2	3	4	
<i>seq</i>	C	O	C	O	\0	ligne 7
<i>overlap</i>	-1	0				

Recherche de motif

Calcul du bord maximal

• Pré-traitement « malin »

Calcul efficace des chevauchements de tous les préfixes

```
1 int* Overlap( const string &seq) {
2     size_t lg = seq.length();
3     size_t* overlap = new int[lg + 1];
4     overlap[0] = (size_t)-1;
5     for (size_t i = 0; i < lg; i++) {
6         overlap[i + 1] = overlap[i] + 1;
7         while (overlap[i + 1] > 0 &&
8                 seq[i] != seq[overlap[i + 1] - 1]) {
9             overlap[i + 1] = overlap[overlap[i + 1] - 1] + 1;
10        }
11    }
12    return overlap;
13 }
```

i	0	1	2	3	4	
seq	C	O	C	O	\0	ligne 5
$overlap$	-1	0				

Recherche de motif

Calcul du bord maximal



● Pré-traitement « malin »

Calcul efficace des chevauchements de tous les préfixes

```
1 int* Overlap( const string &seq) {
2     size_t lg = seq.length();
3     size_t* overlap = new int[lg + 1];
4     overlap[0] = (size_t)-1;
5     for (size_t i = 0; i < lg; i++) {
6         overlap[i + 1] = overlap[i] + 1;
7         while (overlap[i + 1] > 0 &&
8                 seq[i] != seq[overlap[i + 1] - 1]) {
9             overlap[i + 1] = overlap[overlap[i + 1] - 1] + 1;
10        }
11    }
12    return overlap;
13 }
```

Alignement visuel de séquences

Recherche de motif

Algorithm naïf

Illustration

Algorithm KMP

Calcul du bord maximal

Extraction de motif

i	0	1	2	3	4	
seq	C	O	C	O	\0	ligne 6
$overlap$	-1	0	1			

Recherche de motif

Calcul du bord maximal

- Pré-traitement « malin »

Calcul efficace des chevauchements de tous les préfixes

```

1 int* Overlap( const string &seq) {
2     size_t lg = seq.length();
3     size_t* overlap = new int[lg + 1];
4     overlap[0] = (size_t)-1;
5     for (size_t i = 0; i < lg; i++) {
6         overlap[i + 1] = overlap[i] + 1;
7         while (overlap[i + 1] > 0 &&
8                 seq[i] != seq[overlap[i + 1] - 1]) {
9             overlap[i + 1] = overlap[overlap[i + 1] - 1] + 1;
10        }
11    }
12    return overlap;
13 }
```

<i>i</i>	0	1	2	3	4	
<i>seq</i>	C	O	C	O	\0	ligne 7
<i>overlap</i>	-1	0	1			

Recherche de motif

Calcul du bord maximal



● Pré-traitement « malin »

Calcul efficace des chevauchements de tous les préfixes

```
1 int* Overlap( const string &seq) {
2     size_t lg = seq.length();
3     size_t* overlap = new int[lg + 1];
4     overlap[0] = (size_t)-1;
5     for (size_t i = 0; i < lg; i++) {
6         overlap[i + 1] = overlap[i] + 1;
7         while (overlap[i + 1] > 0 &&
8                 seq[i] != seq[overlap[i + 1] - 1]) {
9             overlap[i + 1] = overlap[overlap[i + 1] - 1] + 1;
10        }
11    }
12    return overlap;
13 }
```

Alignement visuel de séquences

Recherche de motif

Algorithm naïf

Illustration

Algorithm KMP

Calcul du bord maximal

Extraction de motif

i	0	1	2	3	4	
seq	C	O	C	O	\0	ligne 8
$overlap$	-1	0	1			

Recherche de motif

Calcul du bord maximal

- Pré-traitement « malin »

Calcul efficace des chevauchements de tous les préfixes

```

1 int* Overlap( const string &seq) {
2     size_t lg = seq.length();
3     size_t* overlap = new int[lg + 1];
4     overlap[0] = (size_t)-1;
5     for (size_t i = 0; i < lg; i++) {
6         overlap[i + 1] = overlap[i] + 1;
7         while (overlap[i + 1] > 0 &&
8                 seq[i] != seq[overlap[i + 1] - 1]) {
9             overlap[i + 1] = overlap[overlap[i + 1] - 1] + 1;
10        }
11    }
12    return overlap;
13 }
```

i	0	1	2	3	4	
seq	C	O	C	O	\0	ligne 9
$overlap$	-1	0	0			

Recherche de motif

Calcul du bord maximal

- Pré-traitement « malin »

Calcul efficace des chevauchements de tous les préfixes

```

1 int* Overlap( const string &seq) {
2     size_t lg = seq.length();
3     size_t* overlap = new int[lg + 1];
4     overlap[0] = (size_t)-1;
5     for (size_t i = 0; i < lg; i++) {
6         overlap[i + 1] = overlap[i] + 1;
7         while (overlap[i + 1] > 0 &&
8                 seq[i] != seq[overlap[i + 1] - 1]) {
9             overlap[i + 1] = overlap[overlap[i + 1] - 1] + 1;
10        }
11    }
12    return overlap;
13 }
```

i	0	1	2	3	4	
seq	C	O	C	O	\0	ligne 7
$overlap$	-1	0	0			

Recherche de motif

Calcul du bord maximal

- Pré-traitement « malin »

Calcul efficace des chevauchements de tous les préfixes

```

1 int* Overlap( const string &seq) {
2     size_t lg = seq.length();
3     size_t* overlap = new int[lg + 1];
4     overlap[0] = (size_t)-1;
5     for (size_t i = 0; i < lg; i++) {
6         overlap[i + 1] = overlap[i] + 1;
7         while (overlap[i + 1] > 0 &&
8                 seq[i] != seq[overlap[i + 1] - 1]) {
9             overlap[i + 1] = overlap[overlap[i + 1] - 1] + 1;
10        }
11    }
12    return overlap;
13 }
```

<i>i</i>	0	1	2	3	4	
<i>seq</i>	C	O	C	O	\0	ligne 5
<i>overlap</i>	-1	0	0			

Recherche de motif

Calcul du bord maximal

- Pré-traitement « malin »

Calcul efficace des chevauchements de tous les préfixes

```

1 int* Overlap( const string &seq) {
2     size_t lg = seq.length();
3     size_t* overlap = new int[lg + 1];
4     overlap[0] = (size_t)-1;
5     for (size_t i = 0; i < lg; i++) {
6         overlap[i + 1] = overlap[i] + 1;
7         while (overlap[i + 1] > 0 &&
8                 seq[i] != seq[overlap[i + 1] - 1]) {
9             overlap[i + 1] = overlap[overlap[i + 1] - 1] + 1;
10        }
11    }
12    return overlap;
13 }
```

i	0	1	2	3	4	
seq	C	O	C	O	\0	ligne 6
$overlap$	-1	0	0	1		

Recherche de motif

Calcul du bord maximal

- Pré-traitement « malin »

Calcul efficace des chevauchements de tous les préfixes

```

1 int* Overlap( const string &seq) {
2     size_t lg = seq.length();
3     size_t* overlap = new int[lg + 1];
4     overlap[0] = (size_t)-1;
5     for (size_t i = 0; i < lg; i++) {
6         overlap[i + 1] = overlap[i] + 1;
7         while (overlap[i + 1] > 0 &&
8                 seq[i] != seq[overlap[i + 1] - 1]) {
9             overlap[i + 1] = overlap[overlap[i + 1] - 1] + 1;
10        }
11    }
12    return overlap;
13 }
```

<i>i</i>	0	1	2	3	4	
<i>seq</i>	C	O	C	O	\0	ligne 7
<i>overlap</i>	-1	0	0	1		

Recherche de motif

Calcul du bord maximal

- Pré-traitement « malin »

Calcul efficace des chevauchements de tous les préfixes

```

1 int* Overlap( const string &seq) {
2     size_t lg = seq.length();
3     size_t* overlap = new int[lg + 1];
4     overlap[0] = (size_t)-1;
5     for (size_t i = 0; i < lg; i++) {
6         overlap[i + 1] = overlap[i] + 1;
7         while (overlap[i + 1] > 0 &&
8                 seq[i] != seq[overlap[i + 1] - 1]) {
9             overlap[i + 1] = overlap[overlap[i + 1] - 1] + 1;
10        }
11    }
12    return overlap;
13 }
```

i	0	1	2	3	4	
seq	C	O	C	O	\O	ligne 8
$overlap$	-1	0	0	1		

Recherche de motif

Calcul du bord maximal

- Pré-traitement « malin »

Calcul efficace des chevauchements de tous les préfixes

```

1 int* Overlap( const string &seq) {
2     size_t lg = seq.length();
3     size_t* overlap = new int[lg + 1];
4     overlap[0] = (size_t)-1;
5     for (size_t i = 0; i < lg; i++) {
6         overlap[i + 1] = overlap[i] + 1;
7         while (overlap[i + 1] > 0 &&
8                 seq[i] != seq[overlap[i + 1] - 1]) {
9             overlap[i + 1] = overlap[overlap[i + 1] - 1] + 1;
10        }
11    }
12    return overlap;
13 }
```

<i>i</i>	0	1	2	3	4	
<i>seq</i>	C	O	C	O	\0	ligne 5
<i>overlap</i>	-1	0	0	1		

Recherche de motif

Calcul du bord maximal

- Pré-traitement « malin »

Calcul efficace des chevauchements de tous les préfixes

```

1 int* Overlap( const string &seq) {
2     size_t lg = seq.length();
3     size_t* overlap = new int[lg + 1];
4     overlap[0] = (size_t)-1;
5     for (size_t i = 0; i < lg; i++) {
6         overlap[i + 1] = overlap[i] + 1;
7         while (overlap[i + 1] > 0 &&
8                 seq[i] != seq[overlap[i + 1] - 1]) {
9             overlap[i + 1] = overlap[overlap[i + 1] - 1] + 1;
10        }
11    }
12    return overlap;
13 }
```

i	0	1	2	3	4	
seq	C	O	C	O	\O	ligne 6
$overlap$	-1	0	0	1	2	

Recherche de motif

Calcul du bord maximal

- Pré-traitement « malin »

Calcul efficace des chevauchements de tous les préfixes

```

1 int* Overlap( const string &seq) {
2     size_t lg = seq.length();
3     size_t* overlap = new int[lg + 1];
4     overlap[0] = (size_t)-1;
5     for (size_t i = 0; i < lg; i++) {
6         overlap[i + 1] = overlap[i] + 1;
7         while (overlap[i + 1] > 0 &&
8                 seq[i] != seq[overlap[i + 1] - 1]) {
9             overlap[i + 1] = overlap[overlap[i + 1] - 1] + 1;
10        }
11    }
12    return overlap;
13 }
```

<i>i</i>	0	1	2	3	4	
<i>seq</i>	C	O	C	O	\0	ligne 7
<i>overlap</i>	-1	0	0	1	2	

Recherche de motif

Calcul du bord maximal

- Pré-traitement « malin »

Calcul efficace des chevauchements de tous les préfixes

```

1 int* Overlap( const string &seq) {
2     size_t lg = seq.length();
3     size_t* overlap = new int[lg + 1];
4     overlap[0] = (size_t)-1;
5     for (size_t i = 0; i < lg; i++) {
6         overlap[i + 1] = overlap[i] + 1;
7         while (overlap[i + 1] > 0 &&
8                 seq[i] != seq[overlap[i + 1] - 1]) {
9             overlap[i + 1] = overlap[overlap[i + 1] - 1] + 1;
10        }
11    }
12    return overlap;
13 }
```

i	0	1	2	3	4	
seq	C	O	C	O	\0	ligne 8
$overlap$	-1	0	0	1	2	

Recherche de motif

Calcul du bord maximal

- Pré-traitement « malin »

Calcul efficace des chevauchements de tous les préfixes

```

1 int* Overlap( const string &seq) {
2     size_t lg = seq.length();
3     size_t* overlap = new int[lg + 1];
4     overlap[0] = (size_t)-1;
5     for (size_t i = 0; i < lg; i++) {
6         overlap[i + 1] = overlap[i] + 1;
7         while (overlap[i + 1] > 0 &&
8                 seq[i] != seq[overlap[i + 1] - 1]) {
9             overlap[i + 1] = overlap[overlap[i + 1] - 1] + 1;
10        }
11    }
12    return overlap;
13 }
```

<i>i</i>	0	1	2	3	4
<i>seq</i>	C	O	C	O	\0
<i>overlap</i>	-1	0	0	1	2

ligne 12

Recherche de motif

Calcul du bord maximal



• Pré-traitement « malin »

Calcul efficace des chevauchements de tous les préfixes

```
1 int* Overlap( const string &seq ) {  
2     size_t* overlap = new size_t[seq.length()];  
3     size_t* Overlap(seq) = ℓ;  
4     overlap[0] = (size_t)-1;  
5     for (size_t i = 0; i < lg; i++) {  
6         seq [i + 1] = overlap[i + 1];  
7         while (overlap[i + 1] > 0 &&  
8                 seq[i] != seq[overlap[i + 1] - 1]) {  
9             overlap[i + 1] = overlap[overlap[i + 1] - 1] + 1;  
10        }  
11    }  
12    return overlap;  
13 }
```

i	0	1	2	3	4	
seq	C	O	C	O	\0	ligne 12
$overlap$	-1	0	0	1	2	

Recherche de motif

Calcul du bord maximal



• Pré-traitement « malin »

Calcul efficace des chevauchements de tous les préfixes

```
1 int* Overlap( const string &seq ) {
2     size_t* lg = &seq.length();
3     size_t* overlap; int i;
4     overlap[0] = -1;
5     for (size_t i = 0; i < *lg; i++) {
6         seq[i];
7         while (overlap[i + 1] > 0 &&
8               seq[i] != seq[overlap[i + 1] - 1]) {
9             overlap[i + 1] = overlap[overlap[i + 1] - 1] +
10            };
11     }
12     return overlap;
13 }
```

i	0	1	2	3	4
seq	C	O	C	O	\0
$overlap$	-1	0	0	1	2

ligne 12

Alignement visuel de séquences

Recherche de motif

Algorithmique naïf

Illustration

Algorithm KMP

Calcul du bord maximal

Extraction de motif

Recherche de motif

Calcul du bord maximal



• Pré-traitement « malin »

Calcul efficace des chevauchements de tous les préfixes

```
1 int* Overlap( const string &seq ) {
2     size_t* lg = &seq.length();
3     size_t* overlap;
4     Overlap(seq.a) > ℓ + 1? ;
5     overlap[0] = (size_t)-1;
6     for (size_t i = 0; i < lg; i++) {
7         seq [ ] ? ;
8         while (Overlap[i + 1] > 0 &&
9                 seq[i] != seq[overlap[i + 1] - 1]) {
10             overlap[i + 1] = overlap[overlap[i + 1] - 1] + ;
11         }
12     }
13 }
```

i	0	1	2	3	4
seq	C	O	C	O	\0
$overlap$	-1	0	0	1	2

ligne 12

Alignement visuel de séquences

Recherche de motif

Algorithm naïf

Illustration

Algorithm KMP

Calcul du bord maximal

Extraction de motif

Recherche de motif

Calcul du bord maximal



• Pré-traitement « malin »

Calcul efficace des chevauchements de tous les préfixes

```
1 int* Overlap( const string &seq ) {
2     size_t lg = seq.length();
3     size_t* overlap = new size_t[lg];
4     overlap[0] = -1;
5     for (size_t i = 0; i < lg; i++) {
6         seq[i] == overlap[i] ? overlap[i + 1] = overlap[i] + 1 : overlap[i + 1] = overlap[i] + 1;
7         while (overlap[i + 1] > 0 && seq[i] != seq[overlap[i + 1] - 1]) {
8             overlap[i + 1] = overlap[i + 1] - 1;
9         }
10    }
11 }
12 return overlap;
13 }
```

i	0	1	2	3	4	
seq	C	O	C	O	\0	
$overlap$	-1	0	0	1	2	ligne 12

Recherche de motif

Calcul du bord maximal



• Pré-traitement « malin »

Calcul efficace des chevauchements de tous les préfixes

```
1 int* Overlap( const string &seq ) {
2     size_t* lg = &seq.length();
3     size_t* overlap;
4     Overlap(seq.a) > l + 1?;
5     overlap[0] = (size_t)-1;
6     for (size_t i = 0; i < lg; i++) {
7         seq [ ] ?;
8         while (Overlap[i + 1] > 0 &&
9             seq[i] != seq[overlap[i + 1] - 1]) {
10            overlap[i + 1] = overlap[overlap[i + 1] - 1] + 1;
11        }
12    }
13 }
```

i	0	1	2	3	4	
seq	C	O	C	O	\0	ligne 12
$overlap$	-1	0	0	1	2	

Recherche de motif

Calcul du bord maximal



• Pré-traitement « malin »

Calcul efficace des chevauchements de tous les préfixes

```
1 int* Overlap( const string &seq ) {
2     size_t* overlap = new size_t[seq.length()];
3     size_t* seq_alpha = seq.c_str();
4     if( Overlap(seq.alpha) > seq.length() - 1 ) {
5         overlap[0] = -1;
6         for( size_t i = 0; i < seq.length(); i++ ) {
7             seq[ i ] = seq_alpha[ i ];
8             while( Overlap[ i + 1 ] > 0 &&
9                   seq[ i ] != seq[ overlap[ i + 1 ] - 1 ] ) {
10                 overlap[ i + 1 ] = overlap[ i + 1 ] + 1;
11             }
12         }
13     }
14 }
```

i	0	1	2	3	4	
seq	C	O	C	O	\0	ligne 12
$overlap$	-1	0	0	1	2	

Recherche de motif

Calcul du bord maximal



• Pré-traitement « malin »

Calcul efficace des chevauchements de tous les préfixes

```
1 int* Overlap( const string &seq ) {  
2     size_t* lg = &seq.length();  
3     size_t* overlap; Overlap(seq.a) > l + 1?;  
4     overlap[0] = (size_t)-1;  
5     for (size_t i = 0; i < lg; i++) {  
6         seq [i] -> [i]; ?  
7         while (Overlap[i + 1] > l &&  
8                 seq[i] != seq[overlap[i + 1] - 1]) {  
9             overlap[i + 1] = Overlap[overlap[i + 1] - 1] +  
10        }  
11    }  
12    return overlap;  
13 }
```

i	0	1	2	3	4	
seq	C	O	C	O	\0	ligne 12
$overlap$	-1	0	0	1	2	

Recherche de motif

Calcul du bord maximal



• Pré-traitement « malin »

Calcul efficace des chevauchements de tous les préfixes

```
1 int* Overlap( const string &seq ) {
2     size_t* lg = &seq.length();
3     size_t* overlap;
4     Overlap(seq.a) > l + 1?;
5     overlap[0] = (size_t)-1;
6     for (size_t i = 0; i < lg; i++) {
7         seq [ ] ?;
8         while (Overlap[i + 1] > 0 &&
9             seq[i] != seq[overlap[i + 1] - 1]) {
10            overlap[i + 1] = overlap[overlap[i + 1] - 1] + ;
11        }
12    }
13 }
```

i	0	1	2	3	4
seq	C	O	C	O	\0
$overlap$	-1	0	0	1	2

ligne 12

Alignement visuel de séquences

Recherche de motif

Algorithmique naïf

Illustration

Algorithmique KMP

Calcul du bord maximal

Extraction de motif

Recherche de motif

Calcul du bord maximal



• Pré-traitement « malin »

Calcul efficace des chevauchements de tous les préfixes

```
1 int* Overlap( const string &seq ) {
2     size_t lg = seq.length();
3     size_t* overlap = new size_t[lg];
4     overlap[0] = -1;
5     for (size_t i = 0; i < lg; i++) {
6         seq[0:i+1] ??
7         while (Overlap[i + 1] <= seq[i]) {
8             if (seq[i] != seq[overlap[i + 1] - 1]) {
9                 overlap[i + 1] = overlap[overlap[i + 1] - 1] + 1;
10            }
11        }
12    }
13 }
```

The diagram shows the string "COCONUT" as a blue horizontal bar with red segments indicating matches and a green segment for the mismatch at index 3. Below it, the overlap array is shown with values -1, 0, 0, 1, 2. A question mark is placed above the first character of the string to indicate the start of the search.

i	0	1	2	3	4	
seq	C	O	C	O	\0	
$overlap$	-1	0	0	1	2	ligne 12

Recherche de motif

Calcul du bord maximal



• Pré-traitement « malin »

Calcul efficace des chevauchements de tous les préfixes

```
1 int* Overlap( const string &seq ) {
2     size_t lg = seq.length();
3     size_t* overlap = new size_t[lg];
4     overlap[0] = -1;
5     for (size_t i = 0; i < lg; i++) {
6         seq[0] = seq[i]; // ?
7         while (seq[i + 1] == seq[overlap[i + 1] + 1]) {
8             if (overlap[i + 1] != -1) {
9                 overlap[i + 1] = overlap[overlap[i + 1] + 1];
10            }
11        }
12    }
13 }
```

i	0	1	2	3	4	
seq	C	O	C	O	\0	ligne 12
$overlap$	-1	0	0	1	2	

Recherche de motif

Calcul du bord maximal



• Pré-traitement « malin »

Calcul efficace des chevauchements de tous les préfixes

```
1 int* Overlap( const string &seq ) {
2     size_t* lg = &seq.length();
3     size_t* overlap;
4     Overlap(seq.a) > ℓ + 1? ;
5     overlap[0] = (size_t)-1;
6     for (size_t i = 0; i < lg; i++) {
7         seq [ ] ; Impossible
8         while (seq[i] != seq[overlap[i + 1] - 1]) {
9             overlap[i + 1] = overlap[overlap[i + 1] - 1] +
10        }
11    }
12    return overlap;
13 }
```

i	0	1	2	3	4
seq	C	O	C	O	\0
$overlap$	-1	0	0	1	2

ligne 12

Alignement visuel de séquences

Recherche de motif

Algorithm naïf

Illustration

Algorithm KMP

Calcul du bord maximal

Extraction de motif

Recherche de motif

Calcul du bord maximal



• Pré-traitement « malin »

Calcul efficace des chevauchements de tous les préfixes

```
1 int* Overlap( const string &seq ) {  
2     size_t* lg = &seq.length();  
3     size_t* overlap; Overlap(seq.alpha) = ℓ + 1;?  
4     overlap[0] = (size_t)-1;  
5     for (size_t i = 0; i < lg; i++) {  
6         seq [i] == seq[i + 1] ? overlap[i + 1] = overlap[i] + 1 : ?;  
7         while (overlap[i + 1] > 0 &&  
8                 seq[i] != seq[overlap[i + 1] - 1]) {  
9             overlap[i + 1] = overlap[overlap[i + 1] - 1] + 1;  
10        }  
11    }  
12    return overlap;  
13 }
```

i	0	1	2	3	4	
seq	C	O	C	O	\0	ligne 12
$overlap$	-1	0	0	1	2	

Recherche de motif

Calcul du bord maximal



• Pré-traitement « malin »

Calcul efficace des chevauchements de tous les préfixes

```
1 int* Overlap( const string &seq ) {  
2     size_t* lg = &seq.length();  
3     size_t* overlap; Overlap(seq.alpha) = ℓ + 1; // ligne 3  
4     overlap[0] = (size_t)-1;  
5     for (size_t i = 0; i < lg; i++) {  
6         seq[i] = seq[i - ℓ]; // ligne 6  
7         while (overlap[i + 1] > 0 &&  
8                 seq[i] != seq[overlap[i + 1] - 1]) {  
9             overlap[i + 1] = overlap[overlap[i + 1] - 1] + 1;  
10        }  
11    }  
12    return overlap;  
13 }
```

Possible

i	0	1	2	3	4	
seq	C	O	C	O	\0	
$overlap$	-1	0	0	1	2	ligne 12

Recherche de motif

Calcul du bord maximal



• Pré-traitement « malin »

Calcul efficace des chevauchements de tous les préfixes

```
1 int* Overlap( const string &seq ) {
2     size_t* lg = &seq.length();
3     size_t* overlap(int*)? i + 1];
4     overlap[0] = (size_t)-1;
5     for (size_t i = 0; i < lg; i++) {
6         seq [i + 1] = overlap[i + 1];
7         while (overlap[i + 1] > 0 &&
8               seq[i] != seq[overlap[i + 1] - 1]) {
9             overlap[i + 1] = overlap[overlap[i + 1] - 1] +
10            };
11        }
12    return overlap;
13 }
```

i	0	1	2	3	4	
seq	C	O	C	O	\0	ligne 12
$overlap$	-1	0	0	1	2	

Recherche de motif

Calcul du bord maximal



• Pré-traitement « malin »

Calcul efficace des chevauchements de tous les préfixes

```
1 int* Overlap( const string &seq ) {
2     size_t lg = seq.length();
3     size_t* overlap = new size_t[lg];
4     overlap[0] = -1;
5     for (size_t i = 0; i < lg; i++) {
6         seq[i];
7         while (overlap[i + 1] >= &seq[i] && overlap[i + 1] - 1) {
8             overlap[i + 1] = overlap[overlap[i + 1] - 1];
9         }
10    }
11    return overlap;
12 }
```

i	0	1	2	3	4	
seq	C	O	C	O	\0	ligne 12
$overlap$	-1	0	0	1	2	

Recherche de motif

Calcul du bord maximal



• Pré-traitement « malin »

Calcul efficace des chevauchements de tous les préfixes

```
1 int* Overlap( const string &seq ) {
2     size_t lg = seq.length();
3     size_t* overlap = new size_t[lg + 1];
4     overlap[0] = -1;
5     for (size_t i = 0; i < lg; i++) {
6         seq[ i + 1 ] = seq[ overlap[i + 1] ];
7         while (overlap[i + 1] > 0 &&
8               seq[i] != seq[overlap[i + 1] - 1]) {
9             overlap[i + 1] = overlap[overlap[i + 1] - 1] +
10            };
11     }
12     return overlap;
13 }
```

i	0	1	2	3	4	
seq	C	O	C	O	\0	ligne 12
$overlap$	-1	0	0	1	2	

Recherche de motif

Calcul du bord maximal



• Pré-traitement « malin »

Calcul efficace des chevauchements de tous les préfixes

```
1 int* Overlap( const string &seq ) {
2     size_t lg = seq.length();
3     size_t* overlap = Overlap(seq);
4     overlap[0] = -1;
5     for (size_t i = 0; i < lg; i++) {
6         seq[i];
7         while (overlap[i + 1] >= i &&
8               seq[i] != seq[overlap[i + 1] - 1]) {
9             overlap[i + 1] = overlap[overlap[i + 1] - 1] + 1;
10        }
11    }
12    return overlap;
13 }
```

i	0	1	2	3	4	
seq	C	O	C	O	\0	
$overlap$	-1	0	0	1	2	ligne 12

Recherche de motif

Calcul du bord maximal



• Pré-traitement « malin »

Calcul efficace des chevauchements de tous les préfixes

```
1 int* Overlap( const string &seq ) {
2     size_t lg = seq.length();
3     size_t* overlap = new size_t[lg];
4     overlap[0] = -1;
5     for (size_t i = 0; i < lg; i++) {
6         seq[i];
7         while (overlap[i + 1] > 0 &&
8               seq[i] == seq[overlap[i + 1] - 1]) {
9             overlap[i + 1] = overlap[overlap[i + 1] - 1] + 1;
10        }
11    }
12    return overlap;
13 }
```

i	0	1	2	3	4	
seq	C	O	C	O	\0	ligne 12
$overlap$	-1	0	0	1	2	

Alignement visuel de séquences

Recherche de motif

Algorithm naïf

Illustration

Algorithm KMP

Calcul du bord maximal

Extraction de motif

Recherche de motif

Calcul du bord maximal



• Pré-traitement « malin »

Calcul efficace des chevauchements de tous les préfixes

```
1 int* Overlap( const string &seq ) {
2     size_t* lg = &seq.length();
3     size_t* overlap(int*){ i + 1];
4     overlap[0] = (size_t)-1;
5     for (size_t i = 0; i < lg; i++) {
6         seq [i] = seq[i];
7         while (overlap[i + 1] > 0 &&
8             seq[i] != seq[overlap[i + 1] - 1]) {
9                 overlap[i + 1] = overlap[overlap[i + 1] - 1] +
10            };
11    }
12    return overlap;
13 }
```

Overlap(seq. α)?

overlap[0] = (size_t)-1;

for (size_t i = 0; i < lg; i++) {

 seq [i] = seq[i];

 while (overlap[i + 1] > 0 &&

 seq[i] != seq[overlap[i + 1] - 1]) {

 overlap[i + 1] = overlap[overlap[i + 1] - 1] +

 };

}

return overlap;

}

i	0	1	2	3	4
seq	C	O	C	O	\0
$overlap$	-1	0	0	1	2

ligne 12

Alignement visuel de séquences

Recherche de motif

Algorithmique naïf

Illustration

Algorithmique KMP

Calcul du bord maximal

Extraction de motif

Recherche de motif

Calcul du bord maximal



• Pré-traitement « malin »

Calcul efficace des chevauchements de tous les préfixes

```
1 int* Overlap( const string &seq ) {
2     size_t* lg = &seq.length();
3     size_t* overlap(int*)? i + 1];
4     overlap[0] = (-size_t) -1;
5     for (size_t i = 0; i < lg; i++) {
6         seq [i + 1] = overlap[i + 1];
7         while (overlap[i + 1] > 0 &&
8             seq[i] != seq[overlap[i + 1] - 1]) {
9                 overlap[i + 1] = overlap[overlap[i + 1] - 1] +
10            }
11    }
12    return overlap;
13 }
```

i	0	1	2	3	4
seq	C	O	C	O	\0
$overlap$	-1	0	0	1	2

ligne 12

Alignement visuel de séquences

Recherche de motif

Algorithm naïf

Illustration

Algorithm KMP

Calcul du bord maximal

Extraction de motif

Recherche de motif

Calcul du bord maximal



• Pré-traitement « malin »

Calcul efficace des chevauchements de tous les préfixes

```
1 int* Overlap( const string &seq ) {  
2     size_t lg = seq.length();  
3     size_t overlap[lg];  
4     overlap[0] = -1;  
5     for (size_t i = 0; i < lg; i++) {  
6         seq [i] = seq[i];  
7         while (overlap[i + 1] > 0 &&  
8                 seq[i] != seq[overlap[i + 1] - 1]) {  
9             overlap[i + 1] = overlap[overlap[i + 1] - 1] + 1;  
10        }  
11    }  
12    return overlap;  
13 }
```

i	0	1	2	3	4	
seq	C	O	C	O	\0	ligne 12
$overlap$	-1	0	0	1	2	



Extraction de motif

Sommaire

HLIN608 – Algorithmique du texte.

Alban MANCHERON

Alignement visuel de séquences

Recherche de motif

Extraction de motif

Contexte

Quelques programmes [presque] naïfs

Algorithme KMR

Structures d'index

III.1/ Alignement visuel de séquences

III.2/ Recherche de motif

III.3/ Extraction de motif

- Contexte
- Quelques programmes [presque] naïfs
- Algorithme KMR
- Structures d'index



Extraction de motif

Contexte

• Recherche *versus* extraction de motif

HLIN608 – Algorithmique du texte.

Alban MANCHERON

Alignement visuel de séquences

Recherche de motif

Extraction de motif

Contexte

Quelques programmes [presque] naïfs

Algorithme KMR

Structures d'index



Extraction de motif

Contexte

- Recherche *versus* extraction de motif
- **Recherche** motif connu *a priori*

HLIN608 – Algorithmique du texte.

Alban MANCHERON

Alignement visuel de séquences

Recherche de motif

Extraction de motif

Contexte

Quelques programmes [presque] naïfs

Algorithme KMR

Structures d'index

Extraction de motif

Contexte



HLIN608 – Algorithmique du texte.

Alban MANCHERON

- Recherche *versus* extraction de motif

Recherche motif connu *a priori*

Extraction motif inconnu *a priori*

Alignement visuel de séquences

Recherche de motif

Extraction de motif

Contexte

Quelques programmes [presque] naïfs

Algorithme KMR

Structures d'index



Extraction de motif

Contexte

- Recherche *versus* extraction de motif

Recherche motif connu *a priori*

Extraction motif inconnu *a priori*

⇒ Espace des possible très différent !!!

HLIN608 – Algorithmique du texte.

Alban MANCHERON

Alignement visuel de séquences

Recherche de motif

Extraction de motif

Contexte

Quelques programmes [presque] naïfs

Algorithme KMR

Structures d'index



Extraction de motif

Contexte

- Recherche *versus* extraction de motif

Recherche motif connu *a priori*

Extraction motif inconnu *a priori*

⇒ Espace des possible très différent !!!

⇒ Incidence de la notion de similarité

HLIN608 – Algorithmique du texte.

Alban MANCHERON

Alignement visuel de séquences

Recherche de motif

Extraction de motif

Contexte

Quelques programmes [presque] naïfs

Algorithme KMR

Structures d'index



Extraction de motif

Contexte

HLIN608 – Algorithmique du texte.

Alban MANCHERON

Alignement visuel de séquences

Recherche de motif

Extraction de motif

Contexte

Quelques programmes [presque] naïfs

Algorithme KMR

Structures d'index

- Recherche *versus* extraction de motif

Recherche motif connu *a priori*

Extraction motif inconnu *a priori*

⇒ Espace des possible très différent !!!

⇒ Incidence de la notion de similarité

Problème 1 Extraction des motifs répétés de longueur k



Extraction de motif

Contexte

HLIN608 – Algorithmique du texte.

Alban MANCHERON

- Recherche *versus* extraction de motif

Recherche motif connu *a priori*

Extraction motif inconnu *a priori*

⇒ Espace des possible très différent !!!

⇒ Incidence de la notion de similarité

Problème 1 Extraction des motifs répétés de longueur k

Problème 2 Extraction du(des) plus long(s) motif(s) répété(s)

Alignement visuel de séquences

Recherche de motif

Extraction de motif

Contexte

Quelques programmes [presque] naïfs

Algorithme KMR

Structures d'index

Extraction de motif

Quelques programmes [presque] naïfs

Extraction exacte

```
1  bool Extract(const string &seq, unsigned int k) {
2      unsigned int i, j, l, lg = seq.length();
3      bool result = false;
4      for (i = 0; i < lg - k; i++) {
5          string motif = seq.substr(i, k);
6          int *overlap = Overlap(motif);
7          string msg = "";
8          j = i + 1;
9          while (j < lg - k) {
10              bool ok = true;
11              for (l = 0; ok && (l < k); l++) {
12                  ok = (seq[j + l] == motif[l]);
13              }
14              if (ok) {
15                  if (msg.empty()) {
16                      msg = string("Le motif ") + motif + " apparaît aux positions " + i;
17                  }
18                  msg.append(", ").append(j);
19                  l--;
20              }
21              j += l - overlap[l + 1] + 1;
22          }
23          delete [] overlap;
24          if (!msg.empty()) {
25              ostream << msg << endl;
26              result = true;
27          }
28      }
29      return result;
30 }
```

HLIN608 – Algorithmique du texte.

Alban MANCHERON

Alignement visuel de séquences

Recherche de motif

Extraction de motif

Contexte

Quelques programmes [presque] naïfs

Algorithmme KMR

Structures d'index

Extraction de motif

Quelques programmes [presque] naïfs

Extraction exacte

```
1  bool Extract(const string &seq, unsigned int k) {
2      unsigned int i, j, l, lg = seq.length();
3      bool result = false;
4      for (i = 0; i < lg - k; i++) {
5          string motif = seq.substr(i, k);
6          int *overlap = Overlap(motif);
7          string msg = "";
8          j = i + 1;
9          while (j < lg - k) {
10              bool ok = true;
11              for (l = 0; ok && (l < k); l++) {
12                  ok = (seq[j + l] == motif[l]);
13              }
14              if (ok) {
15                  if (msg.empty()) {
16                      msg = string("Le motif ") + motif + " apparaît aux positions " + i;
17                  }
18                  msg.append(", ").append(j);
19                  l--;
20              }
21              j += l - overlap[l + 1] + 1;
22          }
23          delete [] overlap;
24          if (!msg.empty()) {
25              ostream << msg << endl;
26              result = true;
27          }
28      }
29      return result;
30 }
```

Complexité en $O(n^2 - k^2)$

HLIN608 – Algorithmique du texte.

Alban MANCHERON

Alignement visuel de séquences

Recherche de motif

Extraction de motif

Contexte

Quelques programmes [presque] naïfs

Algorithmme KMR

Structures d'index



Extraction de motif

Quelques programmes [presque] naïfs

HLIN608 – Algorithmique du texte.

Alban MANCHERON

Alignement visuel de séquences

Recherche de motif

Extraction de motif

Contexte

Quelques programmes [presque] naïfs

Algorithme KMR

Structures d'index

Extraction exacte

```
1 int Extract(const string &seq) {  
2     unsigned int i, j, k, l;  
3     for (k = seq.length() - 1; k > 0; k--) {  
4         if (Extract(seq, k)) {  
5             return k;  
6         }  
7     }  
8     return 0;  
9 }
```



Extraction de motif

Quelques programmes [presque] naïfs

HLIN608 – Algorithmique du texte.

Alban MANCHERON

Alignement visuel de séquences

Recherche de motif

Extraction de motif

Contexte

Quelques programmes [presque] naïfs

Algorithme KMR

Structures d'index

Extraction exacte

```
1 int Extract(const string &seq) {  
2     unsigned int i, j, k, l;  
3     for (k = seq.length() - 1; k > 0; k--) {  
4         if (Extract(seq, k)) {  
5             return k;  
6         }  
7     }  
8     return 0;  
9 }
```

Complexité en $\begin{cases} \Omega(n) \\ O(n^3) \end{cases}$

Extraction de motif

Algorithme KMR [1/3]



⇒ KARP, MILLER & ROSENBERG, 1972

HLIN608 – Algorithmique du texte.

Alban MANCHERON

Alignement visuel de séquences

Recherche de motif

Extraction de motif

Contexte

Quelques programmes [presque] naïfs

Algorithme KMR

Structures d'index



Extraction de motif

Algorithme KMR [1/3]

⇒ KARP, MILLER & ROSENBERG, 1972

Définition (Relation d'équivalence)

Deux positions i et j dans une chaîne de caractères s de longueur n sont k -équivalentes, noté $i \ E_k \ j$, si et seulement si les deux facteurs de longueur k débutant aux positions i et j dans s sont identiques ($s[i..i + k - 1] = s[j..j + k - 1]$).

HLIN608 – Algorithmique du texte.

Alban MANCHERON

Alignement visuel de séquences

Recherche de motif

Extraction de motif

Contexte

Quelques programmes [presque] naïfs

Algorithme KMR

Structures d'index

Extraction de motif

Algorithme KMR [1/3]

⇒ KARP, MILLER & ROSENBERG, 1972

Définition (Relation d'équivalence)

Deux positions i et j dans une chaîne de caractères s de longueur n sont k -équivalentes, noté $i \ E_k j$, si et seulement si les deux facteurs de longueur k débutant aux positions i et j dans s sont identiques ($s[i..i + k - 1] = s[j..j + k - 1]$).

Lemme

Soient a, b, i et j tels que $1 \leq i \leq n$, $1 \leq j \leq n$, $a \geq b > 0$ et $\max(i, j) + a + b - 1 \leq n$, alors

$i \ E_{a+b} j \Leftrightarrow (i \ E_a j \wedge i + b \ E_a j + b)$, avec $b \leq a$.

HLIN608 – Algorithmique du texte.

Alban MANCHERON

Alignement visuel de séquences

Recherche de motif

Extraction de motif

Contexte

Quelques programmes [presque] naïfs

Algorithme KMR

Structures d'index

Extraction de motif

Algorithme KMR [1/3]

⇒ KARP, MILLER & ROSENBERG, 1972

Définition (Relation d'équivalence)

Deux positions i et j dans une chaîne de caractères s de longueur n sont k -équivalentes, noté $i \ E_k j$, si et seulement si les deux facteurs de longueur k débutant aux positions i et j dans s sont identiques ($s[i..i + k - 1] = s[j..j + k - 1]$).

Lemme

Soient a, b, i et j tels que $1 \leq i \leq n$, $1 \leq j \leq n$, $a \geq b > 0$ et $\max(i, j) + a + b - 1 \leq n$, alors

$i \ E_{a+b} j \Leftrightarrow (i \ E_a j \wedge i + b \ E_a j + b)$, avec $b \leq a$.

s

HLIN608 – Algorithmique du texte.

Alban MANCHERON

Alignement visuel de séquences

Recherche de motif

Extraction de motif

Contexte

Quelques programmes [presque] naïfs

Algorithme KMR

Structures d'index

Extraction de motif

Algorithme KMR [1/3]

⇒ KARP, MILLER & ROSENBERG, 1972

Définition (Relation d'équivalence)

Deux positions i et j dans une chaîne de caractères s de longueur n sont k -équivalentes, noté $i E_k j$, si et seulement si les deux facteurs de longueur k débutant aux positions i et j dans s sont identiques ($s[i..i + k - 1] = s[j..j + k - 1]$).

Lemme

Soient a, b, i et j tels que $1 \leq i \leq n$, $1 \leq j \leq n$, $a \geq b > 0$ et $\max(i, j) + a + b - 1 \leq n$, alors

$i E_{a+b} j \Leftrightarrow (i E_a j \wedge i + b E_a j + b)$, avec $b \leq a$.



HLIN608 – Algorithmique du texte.

Alban MANCHERON

Alignement visuel de séquences

Recherche de motif

Extraction de motif

Contexte

Quelques programmes [presque] naïfs

Algorithme KMR

Structures d'index

Extraction de motif

Algorithme KMR [1/3]

⇒ KARP, MILLER & ROSENBERG, 1972

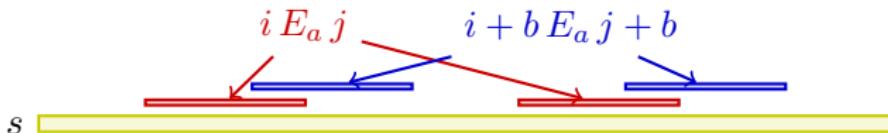
Définition (Relation d'équivalence)

Deux positions i et j dans une chaîne de caractères s de longueur n sont k -équivalentes, noté $i E_k j$, si et seulement si les deux facteurs de longueur k débutant aux positions i et j dans s sont identiques ($s[i..i + k - 1] = s[j..j + k - 1]$).

Lemme

Soient a, b, i et j tels que $1 \leq i \leq n$, $1 \leq j \leq n$, $a \geq b > 0$ et $\max(i, j) + a + b - 1 \leq n$, alors

$i E_{a+b} j \Leftrightarrow (i E_a j \wedge i + b E_a j + b)$, avec $b \leq a$.



HLIN608 – Algorithmique du texte.

Alban MANCHERON

Alignement visuel de séquences

Recherche de motif

Extraction de motif

Contexte

Quelques programmes [presque] naïfs

Algorithme KMR

Structures d'index

Extraction de motif

Algorithme KMR [1/3]

⇒ KARP, MILLER & ROSENBERG, 1972

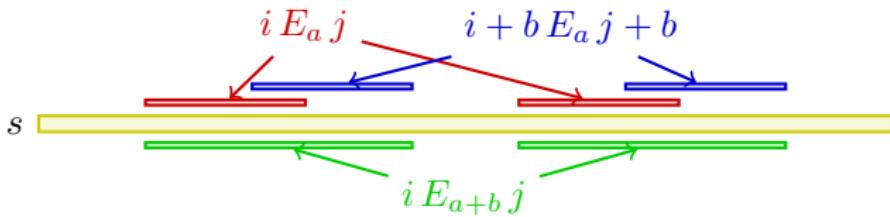
Définition (Relation d'équivalence)

Deux positions i et j dans une chaîne de caractères s de longueur n sont k -équivalentes, noté $i E_k j$, si et seulement si les deux facteurs de longueur k débutant aux positions i et j dans s sont identiques ($s[i..i + k - 1] = s[j..j + k - 1]$).

Lemme

Soient a, b, i et j tels que $1 \leq i \leq n$, $1 \leq j \leq n$, $a \geq b > 0$ et $\max(i, j) + a + b - 1 \leq n$, alors

$i E_{a+b} j \Leftrightarrow (i E_a j \wedge i + b E_a j + b)$, avec $b \leq a$.



HLIN608 – Algorithmique du texte.

Alban MANCHERON

Alignement visuel de séquences

Recherche de motif

Extraction de motif

Contexte

Quelques programmes [presque] naïfs

Algorithme KMR

Structures d'index



Extraction de motif

Algorithme KMR [2/3]

Algorithme KMR \Rightarrow construction des relations E_k à partir de E_1

HLIN608 – Algorithmique du texte.

Alban MANCHERON

Alignement visuel de séquences

Recherche de motif

Extraction de motif

Contexte

Quelques programmes [presque] naïfs

Algorithme KMR

Structures d'index



Extraction de motif

Algorithme KMR [2/3]

Algorithme KMR \Rightarrow construction des relations E_k à partir de E_1

1 Principe de fonctionnement

HLIN608 – Algorithmique du texte.

Alban MANCHERON

Alignement visuel de séquences

Recherche de motif

Extraction de motif

Contexte

Quelques programmes [presque] naïfs

Algorithme KMR

Structures d'index



Extraction de motif

Algorithme KMR [2/3]

Algorithme KMR \Rightarrow construction des relations E_k à partir de E_1

1 Principe de fonctionnement

- 1 La relation E_k est représentée par le vecteur v_k de taille $n - k + 1$

HLIN608 – Algorithmique du texte.

Alban MANCHERON

Alignement visuel de séquences

Recherche de motif

Extraction de motif

Contexte

Quelques programmes [presque] naïfs

Algorithme KMR

Structures d'index



Extraction de motif

Algorithme KMR [2/3]

Algorithme KMR \Rightarrow construction des relations E_k à partir de E_1

1 Principe de fonctionnement

- 1 La relation E_k est représentée par le vecteur v_k de taille $n - k + 1$
- 2 Chaque relation E_k détermine e_k classes, numérotées de 1 à e_k .

HLIN608 – Algorithmique du texte.

Alban MANCHERON

Alignement visuel de séquences

Recherche de motif

Extraction de motif

Contexte

Quelques programmes [presque] naïfs

Algorithme KMR

Structures d'index



Extraction de motif

Algorithme KMR [2/3]

Algorithme KMR \Rightarrow construction des relations E_k à partir de E_1

1 Principe de fonctionnement

- 1 La relation E_k est représentée par le vecteur v_k de taille $n - k + 1$
- 2 Chaque relation E_k détermine e_k classes, numérotées de 1 à e_k .
- 3 E_{a+b} est construit à partir de E_a .

HLIN608 – Algorithmique du texte.

Alban MANCHERON

Alignement visuel de séquences

Recherche de motif

Extraction de motif

Contexte

Quelques programmes [presque] naïfs

Algorithme KMR

Structures d'index



Extraction de motif

Algorithme KMR [2/3]

Algorithme KMR \Rightarrow construction des relations E_k à partir de E_1

1 Principe de fonctionnement

- 1 La relation E_k est représentée par le vecteur v_k de taille $n - k + 1$
- 2 Chaque relation E_k détermine e_k classes, numérotées de 1 à e_k .
- 3 E_{a+b} est construit à partir de E_a .
- 4 Utilisation de deux ensembles \mathcal{P} et \mathcal{Q} de e_a piles chacun.

HLIN608 – Algorithmique du texte.

Alban MANCHERON

Alignement visuel de séquences

Recherche de motif

Extraction de motif

Contexte

Quelques programmes [presque] naïfs

Algorithme KMR

Structures d'index



Extraction de motif

Algorithme KMR [2/3]

Algorithme KMR \Rightarrow construction des relations E_k à partir de E_1

1 Principe de fonctionnement

- 1 La relation E_k est représentée par le vecteur v_k de taille $n - k + 1$
- 2 Chaque relation E_k détermine e_k classes, numérotées de 1 à e_k .
- 3 E_{a+b} est construit à partir de E_a .
- 4 Utilisation de deux ensembles \mathcal{P} et \mathcal{Q} de e_a piles chacun.

2 Construction

HLIN608 – Algorithmique du texte.

Alban MANCHERON

Alignement visuel de séquences

Recherche de motif

Extraction de motif

Contexte

Quelques programmes [presque] naïfs

Algorithme KMR

Structures d'index

Extraction de motif

Algorithme KMR [2/3]

Algorithme KMR \Rightarrow construction des relations E_k à partir de E_1

1 Principe de fonctionnement

- 1 La relation E_k est représentée par le vecteur v_k de taille $n - k + 1$
- 2 Chaque relation E_k détermine e_k classes, numérotées de 1 à e_k .
- 3 E_{a+b} est construit à partir de E_a .
- 4 Utilisation de deux ensembles \mathcal{P} et \mathcal{Q} de e_a piles chacun.

2 Construction

- 1 Chaque position i de la séquence s est placé dans la pile de l'ensemble \mathcal{P} qui correspond à la classe de E_a du mot $s[i..i+a-1]$ (*i.e.*, $\mathcal{P}_{v_a[i]}$)

HLIN608 – Algorithmique du texte.

Alban MANCHERON

Alignement visuel de séquences

Recherche de motif

Extraction de motif

Contexte

Quelques programmes [presque] naïfs

Algorithme KMR

Structures d'index



Extraction de motif

Algorithme KMR [2/3]

Algorithme KMR \Rightarrow construction des relations E_k à partir de E_1

1 Principe de fonctionnement

- 1 La relation E_k est représentée par le vecteur v_k de taille $n - k + 1$
- 2 Chaque relation E_k détermine e_k classes, numérotées de 1 à e_k .
- 3 E_{a+b} est construit à partir de E_a .
- 4 Utilisation de deux ensembles \mathcal{P} et \mathcal{Q} de e_a piles chacun.

2 Construction

- 1 Chaque position i de la séquence s est placé dans la pile de l'ensemble \mathcal{P} qui correspond à la classe de E_a du mot $s[i..i+a-1]$ (*i.e.*, $\mathcal{P}_{v_a[i]}$)
- 2 Les piles de \mathcal{P} sont dépilerées successivement. Chaque position i est placée dans la pile $\mathcal{Q}_{v_a[i+b]}$.

HLIN608 – Algorithmique du texte.

Alban MANCHERON

Alignement visuel de séquences

Recherche de motif

Extraction de motif

Contexte

Quelques programmes [presque] naïfs

Algorithme KMR

Structures d'index

Extraction de motif

Algorithme KMR [2/3]

Algorithme KMR \Rightarrow construction des relations E_k à partir de E_1

1 Principe de fonctionnement

- 1 La relation E_k est représentée par le vecteur v_k de taille $n - k + 1$
- 2 Chaque relation E_k détermine e_k classes, numérotées de 1 à e_k .
- 3 E_{a+b} est construit à partir de E_a .
- 4 Utilisation de deux ensembles \mathcal{P} et \mathcal{Q} de e_a piles chacun.

2 Construction

- 1 Chaque position i de la séquence s est placé dans la pile de l'ensemble \mathcal{P} qui correspond à la classe de E_a du mot $s[i..i+a-1]$ (*i.e.*, $\mathcal{P}_{v_a[i]}$)
- 2 Les piles de \mathcal{P} sont dépilerées successivement. Chaque position i est placée dans la pile $\mathcal{Q}_{v_a[i+b]}$.
- 3 Les piles de \mathcal{Q} sont dépilerées successivement. La classe $v_a[i]$ de chaque position i déjà retirée est comparée à la classe précédente. Si les deux classes sont différentes, les positions proviennent de deux piles de \mathcal{P} différentes et le nombre de classes de E_{a+b} est incrémenté.



Extraction de motif

Algorithme KMR [3/3]

i	1	2	3	4	5	6	7	8	9
s	R	O	U	D	O	U	D	O	U

HLIN608 – Algorithmique du texte.

Alban MANCHERON

Alignement visuel de séquences

Recherche de motif

Extraction de motif

Contexte

Quelques programmes [presque] naïfs

Algorithme KMR

Structures d'index



Extraction de motif

Algorithme KMR [3/3]

$i \quad 1 \ 2 \ 3 \ 4 \ 5 \ 6 \ 7 \ 8 \ 9$
 $s \text{ ROUDOUDOU}$

$e_1 \Rightarrow ?$

HLIN608 – Algorithmique du texte.

Alban MANCHERON

Alignement visuel de séquences

Recherche de motif

Extraction de motif

Contexte

Quelques programmes [presque] naïfs

Algorithme KMR

Structures d'index

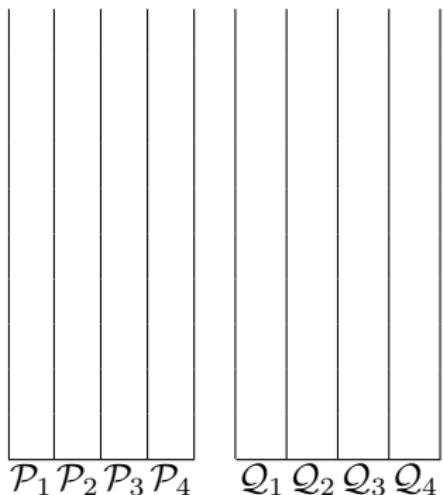


Extraction de motif

Algorithme KMR [3/3]

i	1	2	3	4	5	6	7	8	9
s	R	O	U	D	O	U	D	O	U
v_1	1	2	3	4	2	3	4	2	3

$$\underline{e_1} \Rightarrow \underline{\underline{4}}$$



classe 1 $s[1] = R$

classe 2 $s[2] = s[5] = s[8] = O$

classe 3 $s[3] = s[6] = s[9] = U$

classe 4 $s[4] = s[7] = D$

HLIN608 – Algorithmique du texte.

Alban MANCHERON

Alignement visuel de séquences

Recherche de motif

Extraction de motif

Contexte

Quelques programmes [presque] naïfs

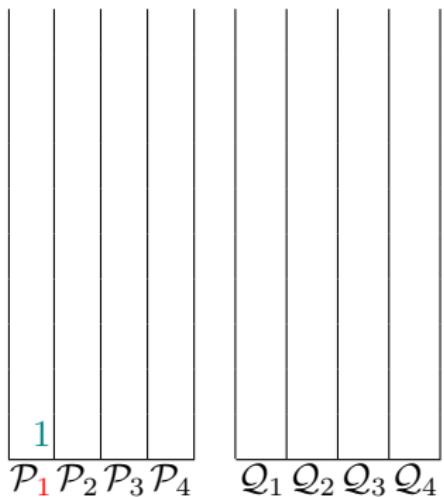
Algorithme KMR

Structures d'index

Extraction de motif

Algorithme KMR [3/3]

$i \quad 1 \ 2 \ 3 \ 4 \ 5 \ 6 \ 7 \ 8 \ 9$
 $s \text{ ROUDOUDOU}$
 $v_1 \textcolor{red}{1} \ 2 \ 3 \ 4 \ 2 \ 3 \ 4 \ 2 \ 3$



$$\underline{e_2} \Rightarrow ?$$
$$a = 1, b = 1$$

classe 1 $s[1] = R$

classe 2 $s[2] = s[5] = s[8] = O$

classe 3 $s[3] = s[6] = s[9] = U$

classe 4 $s[4] = s[7] = D$

HLIN608 – Algorithmique du texte.

Alban MANCHERON

Alignement visuel de séquences

Recherche de motif

Extraction de motif

Contexte

Quelques programmes [presque] naïfs

Algorithme KMR

Structures d'index

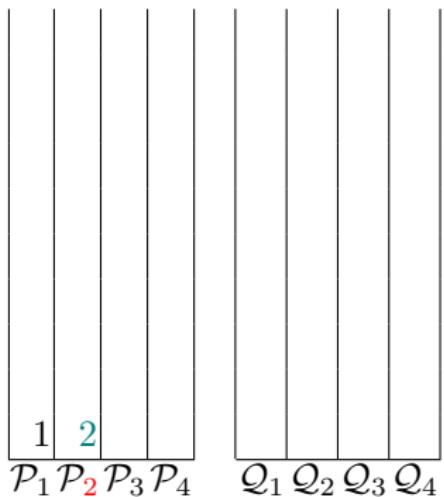
Extraction de motif

Algorithme KMR [3/3]

i	1	2	3	4	5	6	7	8	9
s	R	O	U	D	O	U	D	O	U
v_1	1	2	3	4	2	3	4	2	3

$$e_2 \Rightarrow ?$$

$a = 1, b = 1$



classe 1 $s[1] = R$

classe 2 $s[2] = s[5] = s[8] = O$

classe 3 $s[3] = s[6] = s[9] = U$

classe 4 $s[4] = s[7] = D$

HLIN608 – Algorithmique du texte.

Alban MANCHERON

Alignement visuel de séquences

Recherche de motif

Extraction de motif

Contexte

Quelques programmes [presque] naïfs

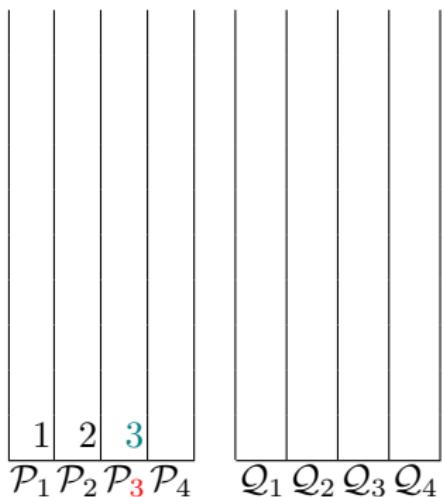
Algorithme KMR

Structures d'index

Extraction de motif

Algorithme KMR [3/3]

i	1	2	3	4	5	6	7	8	9
s	R	O	U	D	O	D	O	U	E
v_1	1	2	3	4	2	3	4	2	3



$$e_2 \Rightarrow ?$$

$a = 1, b = 1$

classe 1 $s[1] = R$

classe 2 $s[2] = s[5] = s[8] = O$

classe 3 $s[3] = s[6] = s[9] = U$

classe 4 $s[4] = s[7] = D$

HLIN608 – Algorithmique du texte.

Alban MANCHERON

Alignement visuel de séquences

Recherche de motif

Extraction de motif

Contexte

Quelques programmes [presque] naïfs

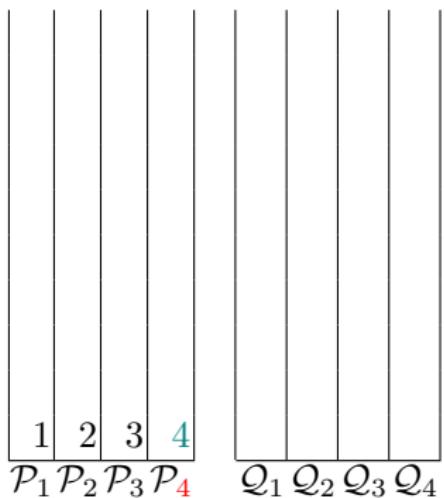
Algorithme KMR

Structures d'index

Extraction de motif

Algorithme KMR [3/3]

i	1	2	3	4	5	6	7	8	9
s	R	O	U	D	O	U	D	O	U
v_1	1	2	3	4	2	3	4	2	3



$$e_2 \Rightarrow ?$$

$a = 1, b = 1$

classe 1 $s[1] = R$

classe 2 $s[2] = s[5] = s[8] = O$

classe 3 $s[3] = s[6] = s[9] = U$

classe 4 $s[4] = s[7] = D$

HLIN608 – Algorithmique du texte.

Alban MANCHERON

Alignement visuel de séquences

Recherche de motif

Extraction de motif

Contexte

Quelques programmes [presque] naïfs

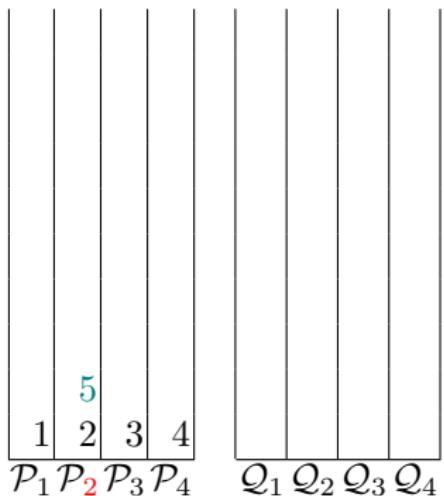
Algorithme KMR

Structures d'index

Extraction de motif

Algorithme KMR [3/3]

i	1	2	3	4	5	6	7	8	9
s	R	O	U	D	O	D	O	U	E
v_1	1	2	3	4	2	3	4	2	3



$$\begin{array}{c} e_2 \Rightarrow ? \\ a = 1, b = 1 \end{array}$$

classe 1 $s[1] = R$

classe 2 $s[2] = s[5] = s[8] = O$

classe 3 $s[3] = s[6] = s[9] = U$

classe 4 $s[4] = s[7] = D$

HLIN608 – Algorithmique du texte.

Alban MANCHERON

Alignement visuel de séquences

Recherche de motif

Extraction de motif

Contexte

Quelques programmes [presque] naïfs

Algorithme KMR

Structures d'index

Extraction de motif

Algorithme KMR [3/3]

i	1	2	3	4	5	6	7	8	9
s	ROUDOUDOU								
v_1	1	2	3	4	2	3	4	2	3

1	2	3	6						
\mathcal{P}_1	\mathcal{P}_2	\mathcal{P}_3	\mathcal{P}_4	\mathcal{Q}_1	\mathcal{Q}_2	\mathcal{Q}_3	\mathcal{Q}_4		

$$\underbrace{e_2}_{a=1} \Rightarrow \underbrace{?}_{b=1}$$

classe 1 $s[1] = R$

classe 2 $s[2] = s[5] = s[8] = O$

classe 3 $s[3] = s[6] = s[9] = U$

classe 4 $s[4] = s[7] = D$

HLIN608 – Algorithmique du texte.

Alban MANCHERON

Alignement visuel de séquences

Recherche de motif

Extraction de motif

Contexte

Quelques programmes [presque] naïfs

Algorithme KMR

Structures d'index

Extraction de motif

Algorithme KMR [3/3]

i	1	2	3	4	5	6	7	8	9
s	R	O	D	O	D	O	U	O	U
v_1	1	2	3	4	2	3	4	2	3

1	5	6	7						
1	2	3	4						
\mathcal{P}_1	\mathcal{P}_2	\mathcal{P}_3	\mathcal{P}_4	\mathcal{Q}_1	\mathcal{Q}_2	\mathcal{Q}_3	\mathcal{Q}_4		

$$\underbrace{e_2}_{a=1} \Rightarrow \underbrace{?}_{b=1}$$

classe 1 $s[1] = R$

classe 2 $s[2] = s[5] = s[8] = O$

classe 3 $s[3] = s[6] = s[9] = U$

classe 4 $s[4] = s[7] = D$

HLIN608 – Algorithmique du texte.

Alban MANCHERON

Alignement visuel de séquences

Recherche de motif

Extraction de motif

Contexte

Quelques programmes [presque] naïfs

Algorithme KMR

Structures d'index

Extraction de motif

Algorithme KMR [3/3]



<i>i</i>	1	2	3	4	5	6	7	8	9
<i>s</i>	ROU	DOU	DOU	OU					
<i>v₁</i>	1	2	3	4	2	3	4	2	3

$$\begin{array}{l} \text{---} \\ e_2 \Rightarrow ? \\ \text{---} \\ a = 1, b = 1 \end{array}$$

classe 1 $s[1] = R$

classe 2 $s[2] = s[5] = s[8] = \circ$

classe 3 $s[3] = s[6] = s[9] = \text{U}$

classe 4 $s[4] = s[7] = D$

Recherche de motif

Extraction de motif

Quelques programmes [presque] naïfs

Extraction de motif

Algorithme KMR [3/3]

$i \quad 1 \ 2 \ 3 \ 4 \ 5 \ 6 \ 7 \ 8 \ 9$
 $s \text{ ROUDOUDOU}$
 $v_1 \ 1 \ 2 \ 3 \ 4 \ 2 \ 3 \ 4 \ 2 \ 3$

8	9								
5	6	7							
1	2	3	4						
\mathcal{P}_1	\mathcal{P}_2	\mathcal{P}_3	\mathcal{P}_4	\mathcal{Q}_1	\mathcal{Q}_2	\mathcal{Q}_3	\mathcal{Q}_4		

$$\underline{e_2} \Rightarrow ?$$

$a = 1, b = 1$

classe 1 $s[1] = R$

classe 2 $s[2] = s[5] = s[8] = O$

classe 3 $s[3] = s[6] = s[9] = U$

classe 4 $s[4] = s[7] = D$

HLIN608 – Algorithmique du texte.

Alban MANCHERON

Alignement visuel de séquences

Recherche de motif

Extraction de motif

Contexte

Quelques programmes [presque] naïfs

Algorithme KMR

Structures d'index



Extraction de motif

Algorithme KMR [3/3]

$i \quad 1 \ 2 \ 3 \ 4 \ 5 \ 6 \ 7 \ 8 \ 9$
 $s \text{ ROUDOUDOU}$
 $v_1 \ 1 \ \textcolor{red}{2} \ 3 \ 4 \ 2 \ 3 \ 4 \ 2 \ 3$

$$\underbrace{e_2}_{a=1} \Rightarrow \underbrace{?}_{b=1}$$

8	9								
5	6	7							
\textcolor{teal}{1}	2	3	4						
\mathcal{P}_1	\mathcal{P}_2	\mathcal{P}_3	\mathcal{P}_4	\mathcal{Q}_1	$\textcolor{red}{Q}_2$	\mathcal{Q}_3	\mathcal{Q}_4		

classe 1 $s[1] = R$

classe 2 $s[2] = s[5] = s[8] = O$

classe 3 $s[3] = s[6] = s[9] = U$

classe 4 $s[4] = s[7] = D$

HLIN608 – Algorithmique du texte.

Alban MANCHERON

Alignement visuel de séquences

Recherche de motif

Extraction de motif

Contexte

Quelques programmes [presque] naïfs

Algorithme KMR

Structures d'index

Extraction de motif

Algorithme KMR [3/3]

$i \quad 1 \ 2 \ 3 \ 4 \ 5 \ 6 \ 7 \ 8 \ 9$
 $s \text{ ROUDOUDOU}$
 $v_1 \ 1 \ 2 \ 3 \ 4 \ 2 \ 3 \ 4 \ 2 \ 3$

8	9								
5	6	7							
2	3	4							
\mathcal{P}_1	\mathcal{P}_2	\mathcal{P}_3	\mathcal{P}_4	\mathcal{Q}_1	\mathcal{Q}_2	\mathcal{Q}_3	\mathcal{Q}_4		

$$\underline{e_2} \Rightarrow ?$$

$$a = 1, b = 1$$

classe 1 $s[1] = R$

classe 2 $s[2] = s[5] = s[8] = O$

classe 3 $s[3] = s[6] = s[9] = U$

classe 4 $s[4] = s[7] = D$

HLIN608 – Algorithmique du texte.

Alban MANCHERON

Alignement visuel de séquences

Recherche de motif

Extraction de motif

Contexte

Quelques programmes [presque] naïfs

Algorithme KMR

Structures d'index

Extraction de motif

Algorithme KMR [3/3]



<i>i</i>	1	2	3	4	5	6	7	8	9
<i>s</i>	ROUDOUUDOU								
<i>v₁</i>	1	2	3	4	2	3	4	2	3

$$\begin{array}{l} \text{---} \\ e_2 \Rightarrow ? \\ \text{---} \\ a = 1, b = 1 \end{array}$$

\mathcal{P}_1	\mathcal{P}_2	\mathcal{P}_3	\mathcal{P}_4				
2	3	4	9				
5	6	7					
				1	8	5	

classe 1 $s[1] = R$

classe 2 $s[2] = s[5] = s[8] = \circ$

classe 3 $s[3] = s[6] = s[9] = \cup$

classe 4 $s[4] = s[7] = D$

Recherche de motif

Extraction de motif

Extraction de motif

Algorithme KMR [3/3]

$i \quad 1 \ 2 \ \textcolor{teal}{3} \ 4 \ 5 \ 6 \ 7 \ 8 \ 9$
 $s \quad \text{ROUDOUDOU}$
 $v_1 \ 1 \ 2 \ \textcolor{red}{3} \ 4 \ 2 \ 3 \ 4 \ 2 \ 3$

9									
6									
7									
$\textcolor{teal}{2}$									
3									
4									
\mathcal{P}_1	\mathcal{P}_2	\mathcal{P}_3	\mathcal{P}_4						
				\mathcal{Q}_1	\mathcal{Q}_2	\mathcal{Q}_3	\mathcal{Q}_4		

$$\underbrace{e_2}_{a=1, b=1} \Rightarrow ?$$

classe 1 $s[1] = \text{R}$

classe 2 $s[2] = s[5] = s[8] = \text{O}$

classe 3 $s[3] = s[6] = s[9] = \text{U}$

classe 4 $s[4] = s[7] = \text{D}$

HLIN608 – Algorithmique du texte.

Alban MANCHERON

Alignement visuel de séquences

Recherche de motif

Extraction de motif

Contexte

Quelques programmes [presque] naïfs

Algorithme KMR

Structures d'index

Extraction de motif

Algorithme KMR [3/3]

i	1	2	3	4	5	6	7	8	9
s	R	O	D	O	D	O	U		
v_1	1	2	3	4	2	3	4	2	3

\mathcal{P}_1	\mathcal{P}_2	\mathcal{P}_3	\mathcal{P}_4						
\mathcal{Q}_1	\mathcal{Q}_2	\mathcal{Q}_3	\mathcal{Q}_4						
9									
6									
7									
3									
4									

$$\underbrace{e_2}_{a=1} \Rightarrow \underbrace{?}_{b=1}$$

classe 1 $s[1] = R$

classe 2 $s[2] = s[5] = s[8] = O$

classe 3 $s[3] = s[6] = s[9] = U$

classe 4 $s[4] = s[7] = D$

HLIN608 – Algorithmique du texte.

Alban MANCHERON

Alignement visuel de séquences

Recherche de motif

Extraction de motif

Contexte

Quelques programmes [presque] naïfs

Algorithme KMR

Structures d'index

Extraction de motif

Algorithme KMR [3/3]



<i>i</i>	1	2	3	4	5	6	7	8	9
<i>s</i>	ROU	DOU	DOU	OU					
<i>v₁</i>	1	2	3	4	2	3	4	2	3

$$\begin{array}{l} \text{---} \\ e_2 \Rightarrow ? \\ \text{---} \\ a = 1, b = 1 \end{array}$$

classe 1 $s[1] = R$

classe 2 $s[2] = s[5] = s[8] = \circ$

classe 3 $s[3] = s[6] = s[9] = \text{U}$

classe 4 $s[4] = s[7] = D$

\mathcal{P}_1	\mathcal{P}_2	\mathcal{P}_3	\mathcal{P}_4	\mathcal{Q}_1	\mathcal{Q}_2	\mathcal{Q}_3	\mathcal{Q}_4
3	6	7	4	1	2	5	8
							6

Recherche de motif

Extraction de motif

Extraction de motif

Algorithme KMR [3/3]



<i>i</i>	1	2	3	4	5	6	7	8	9
<i>s</i>	ROUDOUUDOU								
<i>v₁</i>	1	2	3	4	2	3	4	2	3

$$\begin{array}{l} \text{---} \\ e_2 \Rightarrow ? \\ \text{---} \\ a = 1, b = 1 \end{array}$$

\mathcal{P}_1	\mathcal{P}_2	\mathcal{P}_3	\mathcal{P}_4	7	3
		\mathcal{P}_5	4	6	
				2	
				5	
				8	
				1	
\mathcal{Q}_1	\mathcal{Q}_2	\mathcal{Q}_3	\mathcal{Q}_4		

- classe 1** $s[1] = R$
- classe 2** $s[2] = s[5] = s[8] = O$
- classe 3** $s[3] = s[6] = s[9] = U$
- classe 4** $s[4] = s[7] = D$

Extraction de motif

Algorithme KMR [3/3]



<i>i</i>	1	2	3	4	5	6	7	8	9
<i>s</i>	ROUDOUUDOU								
<i>v₁</i>	1	2	3	4	2	3	4	2	3

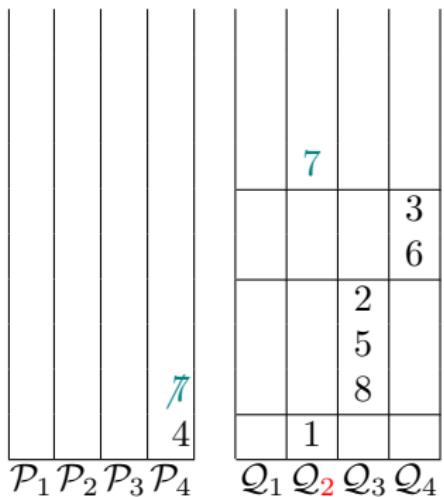
$$\underline{e_2} \Rightarrow ?$$

classe 1 $s[1] = R$

classe 2 $s[2] = s[5] = s[8] = \circ$

classe 3 $s[3] = s[6] = s[9] = \text{U}$

classe 4 $s[4] = s[7] = D$



Recherche de motif

Extraction de motif

Extraction de motif

Algorithme KMR [3/3]



<i>i</i>	1	2	3	4	5	6	7	8	9
<i>s</i>	ROUDOUUDOU								
<i>v₁</i>	1	2	3	4	2	3	4	2	3

$$\begin{array}{l} \text{e}_2 \Rightarrow ? \\ a = 1, b = 1 \end{array}$$

classe 1 $s[1] = R$

classe 2 $s[2] = s[5] = s[8] = \circ$

classe 3 $s[3] = s[6] = s[9] = \text{U}$

classe 4 $s[4] = s[7] = D$

\mathcal{P}_1	\mathcal{P}_2	\mathcal{P}_3	\mathcal{P}_4	$\textcolor{teal}{A}$
\mathcal{Q}_1	\mathcal{Q}_2	$\textcolor{red}{Q}_3$	\mathcal{Q}_4	
4	7	3	6	
		2	5	8
		1		

Alban MANCHERON (EDS)

HJ11N608 – Algorithmique du texte

Année 2017-2018 (v. 1.7)

Extraction de motif

Algorithme KMR [3/3]

i 1 2 3 4 5 6 7 8 9
 s ROUDOUUDOU
 v_1 1 2 3 4 2 3 4 2 3

\mathcal{P}_1	\mathcal{P}_2	\mathcal{P}_3	\mathcal{P}_4						
\mathcal{Q}_1	\mathcal{Q}_2	\mathcal{Q}_3	\mathcal{Q}_4						

$$\underline{e_2} \Rightarrow ?$$

$$a = 1, b = 1$$

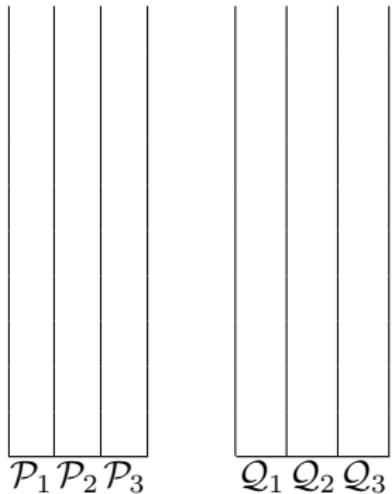
- classe 1** $s[4..5] = s[7..8] = \text{DO}$
- classe 2** $s[3..4] = s[6..7] = \text{UD}$
- classe 3** $s[2..3] = s[5..6] = s[8..9] = \text{OU}$
- classe ignorée** $s[1..2] = \text{RO}$

Extraction de motif

Algorithme KMR [3/3]

i	1	2	3	4	5	6	7	8	9
s	ROUDOUDOU								
v_1	-	3	2	1	3	2	1	3	-

$$\underline{e_2} \Rightarrow \underline{3}$$



classe 1 $s[4..5] = s[7..8] = \text{DO}$

classe 2 $s[3..4] = s[6..7] = \text{UD}$

classe 3

$s[2..3] = s[5..6] = s[8..9] = \text{OU}$

classe ignorée $s[1..2] = \text{RO}$

HLIN608 – Algorithmique du texte.

Alban MANCHERON

Alignement visuel de séquences

Recherche de motif

Extraction de motif

Contexte

Quelques programmes [presque] naïfs

Algorithme KMR

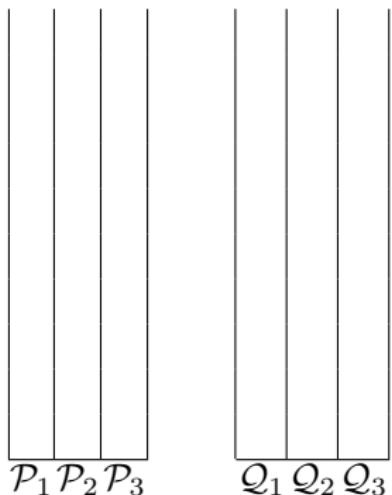
Structures d'index



Extraction de motif

Algorithme KMR [3/3]

$i \quad 1 \ 2 \ 3 \ 4 \ 5 \ 6 \ 7 \ 8 \ 9$
 $s \text{ ROUDOUDOU}$
 $v_2 \text{ -} 3 \ 2 \ 1 \ 3 \ 2 \ 1 \ 3 \text{ -}$



$$\underline{e_4} \Rightarrow \underline{?}$$
$$a = 2, b = 2$$

classe 1 $s[4..5] = s[7..8] = \text{DO}$

classe 2 $s[3..4] = s[6..7] = \text{UD}$

classe 3

$s[2..3] = s[5..6] = s[8..9] = \text{OU}$

classe ignorée $s[1..2] = \text{RO}$

HLIN608 – Algorithmique du texte.

Alban MANCHERON

Alignement visuel de séquences

Recherche de motif

Extraction de motif

Contexte

Quelques programmes [presque] naïfs

Algorithme KMR

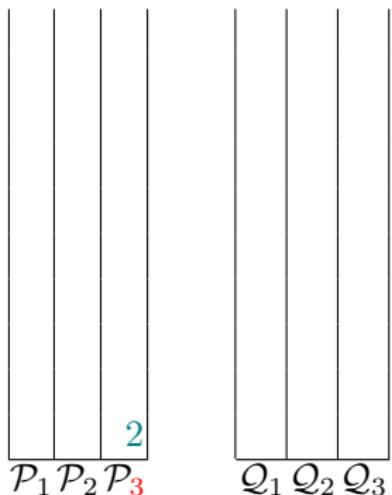
Structures d'index



Extraction de motif

Algorithme KMR [3/3]

$i \quad 1 \ 2 \ 3 \ 4 \ 5 \ 6 \ 7 \ 8 \ 9$
 $s \text{ ROUDOUUDOU}$
 $v_2 - 3 \ 2 \ 1 \ 3 \ 2 \ 1 \ 3 -$



$$\underline{e_4} \Rightarrow ?$$
$$a = 2, b = 2$$

classe 1 $s[4..5] = s[7..8] = \text{DO}$

classe 2 $s[3..4] = s[6..7] = \text{UD}$

classe 3

$s[2..3] = s[5..6] = s[8..9] = \text{OU}$

classe ignorée $s[1..2] = \text{RO}$

HLIN608 – Algorithmique du texte.

Alban MANCHERON

Alignement visuel de séquences

Recherche de motif

Extraction de motif

Contexte

Quelques programmes [presque] naïfs

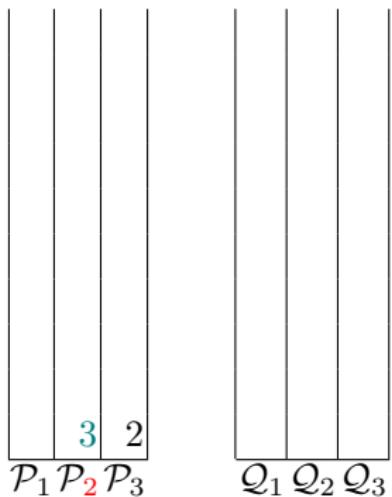
Algorithme KMR

Structures d'index

Extraction de motif

Algorithme KMR [3/3]

$i \quad 1 \ 2 \ \textcolor{teal}{3} \ 4 \ 5 \ 6 \ 7 \ 8 \ 9$
 $s \text{ ROUDOUDOU}$
 $v_2 - 3 \ \textcolor{red}{2} \ 1 \ 3 \ 2 \ 1 \ 3 -$



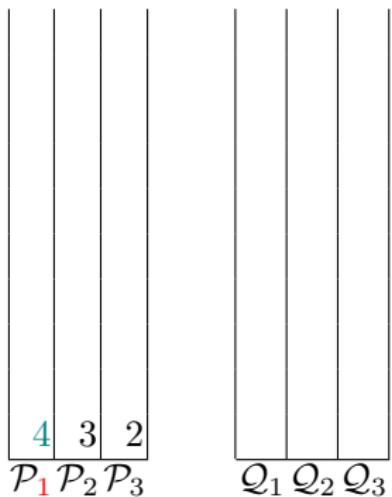
$$\underbrace{e_4}_{a=2,b=2} \Rightarrow ?$$

- classe 1** $s[4..5] = s[7..8] = \text{DO}$
- classe 2** $s[3..4] = s[6..7] = \text{UD}$
- classe 3** $s[2..3] = s[5..6] = s[8..9] = \text{OU}$
- classe ignorée** $s[1..2] = \text{RO}$

Extraction de motif

Algorithme KMR [3/3]

$i \quad 1 \ 2 \ 3 \ \textcolor{teal}{4} \ 5 \ 6 \ 7 \ 8 \ 9$
 $s \text{ ROUDOUDOU}$
 $v_2 - 3 \ 2 \ \textcolor{red}{1} \ 3 \ 2 \ 1 \ 3 -$



$$\underbrace{e_4}_{a=2} \Rightarrow \underbrace{?}_{b=2}$$

classe 1 $s[4..5] = s[7..8] = \text{DO}$

classe 2 $s[3..4] = s[6..7] = \text{UD}$

classe 3

$s[2..3] = s[5..6] = s[8..9] = \text{OU}$

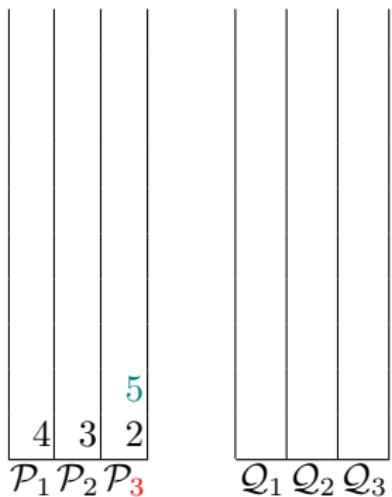
classe ignorée $s[1..2] = \text{RO}$



Extraction de motif

Algorithme KMR [3/3]

$i \quad 1 \ 2 \ 3 \ 4 \ \textcolor{teal}{5} \ 6 \ 7 \ 8 \ 9$
 $s \text{ ROUDOUDOU}$
 $v_2 - 3 \ 2 \ 1 \ \textcolor{red}{3} \ 2 \ 1 \ 3 -$



$$\underbrace{e_4}_{a=2} \Rightarrow \underbrace{?}_{b=2}$$

classe 1 $s[4..5] = s[7..8] = \text{DO}$

classe 2 $s[3..4] = s[6..7] = \text{UD}$

classe 3

$s[2..3] = s[5..6] = s[8..9] = \text{OU}$

classe ignorée $s[1..2] = \text{RO}$

HLIN608 – Algorithmique du texte.

Alban MANCHERON

Alignement visuel de séquences

Recherche de motif

Extraction de motif

Contexte

Quelques programmes [presque] naïfs

Algorithme KMR

Structures d'index

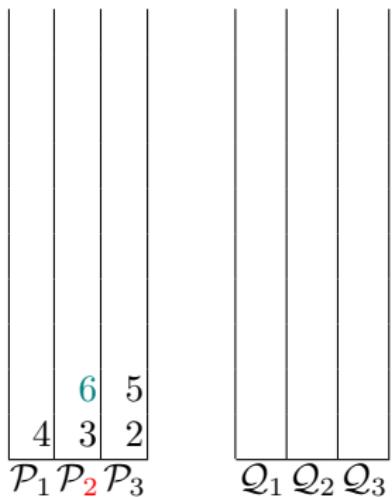
Extraction de motif

Algorithme KMR [3/3]

$i \quad 1 \ 2 \ 3 \ 4 \ 5 \ \textcolor{teal}{6} \ 7 \ 8 \ 9$

$s \quad \text{ROUDOUDOU}$

$v_2 - 3 \ 2 \ 1 \ 3 \ \textcolor{red}{2} \ 1 \ 3 -$



$$\underbrace{e_4}_{a=2} \Rightarrow \underbrace{?}_{b=2}$$

classe 1 $s[4..5] = s[7..8] = \text{DO}$

classe 2 $s[3..4] = s[6..7] = \text{UD}$

classe 3

$s[2..3] = s[5..6] = s[8..9] = \text{OU}$

classe ignorée $s[1..2] = \text{RO}$

HLIN608 – Algorithmique du texte.

Alban MANCHERON

Alignement visuel de séquences

Recherche de motif

Extraction de motif

Contexte

Quelques programmes [presque] naïfs

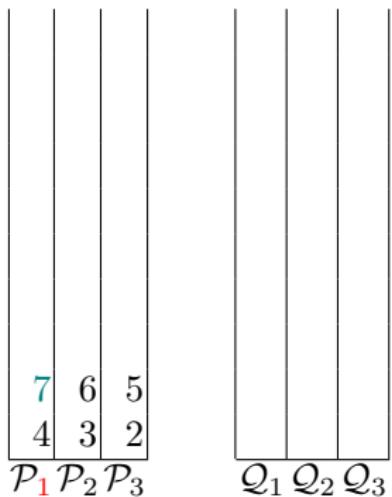
Algorithme KMR

Structures d'index

Extraction de motif

Algorithme KMR [3/3]

$i \quad 1 \ 2 \ 3 \ 4 \ 5 \ 6 \ \textcolor{teal}{7} \ 8 \ 9$
 $s \text{ ROUDOUUDOU}$
 $v_2 - 3 \ 2 \ 1 \ 3 \ 2 \ \textcolor{red}{1} \ 3 -$



$$\underbrace{e_4}_{a=2, b=2} \Rightarrow ?$$

- classe 1** $s[4..5] = s[7..8] = \text{DO}$
- classe 2** $s[3..4] = s[6..7] = \text{UD}$
- classe 3** $s[2..3] = s[5..6] = s[8..9] = \text{OU}$
- classe ignorée** $s[1..2] = \text{RO}$



Extraction de motif

Algorithme KMR [3/3]

$i \quad 1 \ 2 \ 3 \ 4 \ 5 \ 6 \ 7 \ 8 \ 9$
 $s \text{ ROUDOUUDOU}$
 $v_2 - 3 \ 2 \ 1 \ 3 \ 2 \ 1 \ 3 -$

7	6	5	8						
4	3	2							
\mathcal{P}_1	\mathcal{P}_2	\mathcal{P}_3		\mathcal{Q}_1	\mathcal{Q}_2	\mathcal{Q}_3			

$$\underline{e_4} \Rightarrow ?$$
$$a = 2, b = 2$$

classe 1 $s[4..5] = s[7..8] = \text{DO}$

classe 2 $s[3..4] = s[6..7] = \text{UD}$

classe 3

$s[2..3] = s[5..6] = s[8..9] = \text{OU}$

classe ignorée $s[1..2] = \text{RO}$

HLIN608 – Algorithmique du texte.

Alban MANCHERON

Alignement visuel de séquences

Recherche de motif

Extraction de motif

Contexte

Quelques programmes [presque] naïfs

Algorithme KMR

Structures d'index



Extraction de motif

Algorithme KMR [3/3]

$i \quad 1 \ 2 \ 3 \ 4 \ 5 \ 6 \ 7 \ 8 \ 9$

$s \quad \text{ROUDOUDOU}$

$v_2 - 3 \ 2 \ 1 \ 3 \ 2 \ 1 \ 3 -$

7	6	5	8						
4	3	2							

$\mathcal{P}_1 \ \mathcal{P}_2 \ \mathcal{P}_3$ $\mathcal{Q}_1 \ \mathcal{Q}_2 \ \mathcal{Q}_3$

$$\underbrace{e_4}_{a=2} \Rightarrow \underbrace{?}_{b=2}$$

classe 1 $s[4..5] = s[7..8] = \text{DO}$

classe 2 $s[3..4] = s[6..7] = \text{UD}$

classe 3

$s[2..3] = s[5..6] = s[8..9] = \text{OU}$

classe ignorée $s[1..2] = \text{RO}$

HLIN608 – Algorithmique du texte.

Alban MANCHERON

Alignement visuel de séquences

Recherche de motif

Extraction de motif

Contexte

Quelques programmes [presque] naïfs

Algorithme KMR

Structures d'index



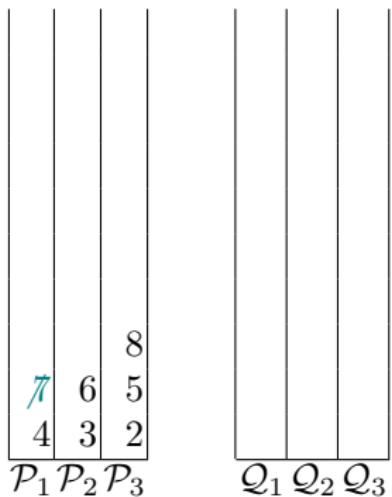
Extraction de motif

Algorithme KMR [3/3]

$i \quad 1 \ 2 \ 3 \ 4 \ 5 \ 6 \ 7 \ 8 \ 9$

$s \quad \text{ROUDOUDOU}$

$v_2 - 3 \ 2 \ 1 \ 3 \ 2 \ 1 \ 3 -$



$$\underbrace{e_4}_{a=2,b=2} \Rightarrow ?$$

classe 1 $s[4..5] = s[7..8] = \text{DO}$

classe 2 $s[3..4] = s[6..7] = \text{UD}$

classe 3

$s[2..3] = s[5..6] = s[8..9] = \text{OU}$

classe ignorée $s[1..2] = \text{RO}$

HLIN608 – Algorithmique du texte.

Alban MANCHERON

Alignement visuel de séquences

Recherche de motif

Extraction de motif

Contexte

Quelques programmes [presque] naïfs

Algorithme KMR

Structures d'index

Extraction de motif

Algorithme KMR [3/3]

<i>i</i>	1	2	3	4	5	6	7	8	9
<i>s</i>	ROU	DOU	DOU	O					
<i>v₂</i>	-	3	2	1	3	2	1	3	-

<i>A</i>	6	5							
	3	2							
<i>P₁</i>	<i>P₂</i>	<i>P₃</i>							
			<i>Q₁</i>	<i>Q₂</i>	<i>Q₃</i>				
				4					

$$\underline{e_4} \Rightarrow ?$$

$a = 2, b = 2$

classe 1 $s[4..5] = s[7..8] = \text{DO}$

classe 2 $s[3..4] = s[6..7] = \text{UD}$

classe 3

$s[2..3] = s[5..6] = s[8..9] = \text{OU}$

classe ignorée $s[1..2] = \text{RO}$

Extraction de motif

Algorithme KMR [3/3]

$i \quad 1 \ 2 \ 3 \ 4 \ 5 \ 6 \ 7 \ 8 \ 9$
 $s \text{ ROUDOUUDOU}$
 $v_2 - 3 \ 2 \ 1 \ 3 \ 2 \ 1 \ 3 -$

\mathcal{P}_1	\mathcal{P}_2	\mathcal{P}_3						
6	5	8						
3	2							

$$\underline{e_4} \Rightarrow ?$$

$$a = 2, b = 2$$

- classe 1** $s[4..5] = s[7..8] = \text{DO}$
- classe 2** $s[3..4] = s[6..7] = \text{UD}$
- classe 3** $s[2..3] = s[5..6] = s[8..9] = \text{OU}$
- classe ignorée** $s[1..2] = \text{RO}$

Extraction de motif

Algorithme KMR [3/3]

i	1	2	3	4	5	6	7	8	9
s	ROU	DOU	DOU	O					
v_2	-	3	2	1	3	2	1	3	-

$$\underbrace{e_4}_{a=2} \Rightarrow \underbrace{?}_{b=2}$$

\mathcal{P}_1	\mathcal{P}_2	\mathcal{P}_3							
8	5	3							
5	2								

\mathcal{Q}_1	\mathcal{Q}_2	\mathcal{Q}_3							
3	6	4							

- classe 1** $s[4..5] = s[7..8] = \text{DO}$
- classe 2** $s[3..4] = s[6..7] = \text{UD}$
- classe 3** $s[2..3] = s[5..6] = s[8..9] = \text{OU}$
- classe ignorée** $s[1..2] = \text{RO}$

Extraction de motif

Algorithme KMR [3/3]



<i>i</i>	1	2	3	4	5	6	7	8	9
<i>s</i>	ROUDOUUDOU								
<i>v</i> ₂	-	3	2	1	3	2	1	3	-

$$\underline{e_4} \Rightarrow ?$$

classe 1 $s[4..5] = s[7..8] = \text{DO}$

classe 2 $s[3..4] = s[6..7] = \text{UD}$

classe 3

$$s[2..3] = s[5..6] = s[8..9] = \text{OU}$$

classe ignorée $s[1..2] = \text{RO}$

Extraction de motif

Algorithme KMR [3/3]

$i \quad 1 \ 2 \ 3 \ 4 \ 5 \ 6 \ \textcolor{teal}{7} \ 8 \ 9$
 $s \quad \text{ROUDOUDOU}$
 $v_2 - 3 \ 2 \ 1 \ 3 \ 2 \ \textcolor{red}{1} \ 3 -$

\mathcal{P}_1	\mathcal{P}_2	\mathcal{P}_3						

\mathcal{Q}_1	\mathcal{Q}_2	\mathcal{Q}_3						

$$\underline{e_4} \Rightarrow ?$$

$$a = 2, b = 2$$

classe 1 $s[4..5] = s[7..8] = \text{DO}$

classe 2 $s[3..4] = s[6..7] = \text{UD}$

classe 3

$s[2..3] = s[5..6] = s[8..9] = \text{OU}$

classe ignorée $s[1..2] = \text{RO}$

5		
		3
		6

Extraction de motif

Algorithme KMR [3/3]



<i>i</i>	1	2	3	4	5	6	7	8	9
<i>s</i>	ROUDOUUDOU								
<i>v</i> ₂	-	3	2	1	3	2	1	3	-

$$\underline{e_4} \Rightarrow ?$$

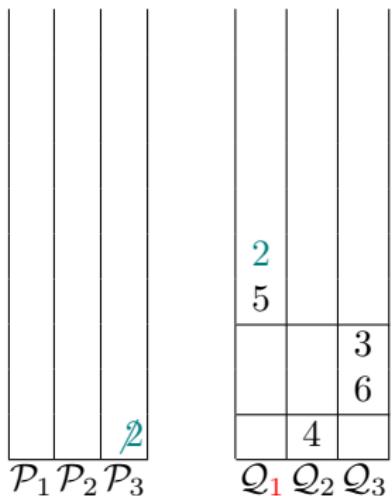
classe 1 $s[4..5] = s[7..8] = \text{DO}$

classe 2 $s[3..4] = s[6..7] = \text{UD}$

classe 3

$$s[2..3] = s[5..6] = s[8..9] = \text{OU}$$

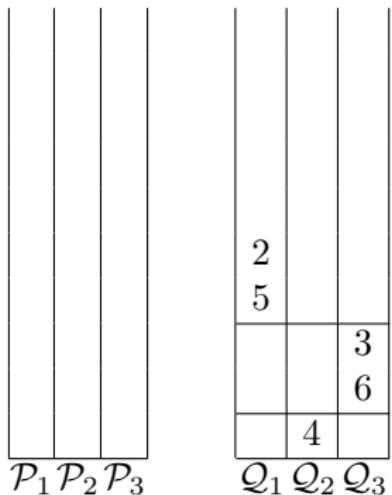
classe ignorée $s[1..2] = \text{RO}$



Extraction de motif

Algorithme KMR [3/3]

$i \quad 1 \ 2 \ 3 \ 4 \ 5 \ 6 \ 7 \ 8 \ 9$
 $s \text{ ROUDOUDOU}$
 $v_2 - 3 \ 2 \ 1 \ 3 \ 2 \ 1 \ 3 -$



$$\underline{e_4} \Rightarrow \underline{?}$$

$$a = 2, b = 2$$

classe 1 $s[2..5] = s[5..8] = \text{OUDO}$
classe 2 $s[3..6] = s[6..9] = \text{UDOU}$
classe ignorée $s[4..7] = \text{DOUD}$

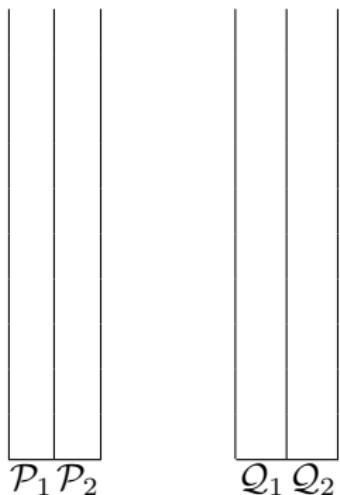
2			3	
5			6	
	4			

Extraction de motif

Algorithme KMR [3/3]

$i \quad 1 \ 2 \ 3 \ 4 \ 5 \ 6 \ 7 \ 8 \ 9$
 $s \text{ ROUDOUDOU}$
 $v_4 - 1 \ 2 - 1 \ 2 - - -$

$\underline{e_4} \Rightarrow 2$



classe 1 $s[2..5] = s[5..8] = \text{OUDO}$
classe 2 $s[3..6] = s[6..9] = \text{UDOU}$
classe ignorée $s[4..7] = \text{DOUD}$

HLIN608 – Algorithmique du texte.

Alban MANCHERON

Alignement visuel de séquences

Recherche de motif

Extraction de motif

Contexte

Quelques programmes [presque] naïfs

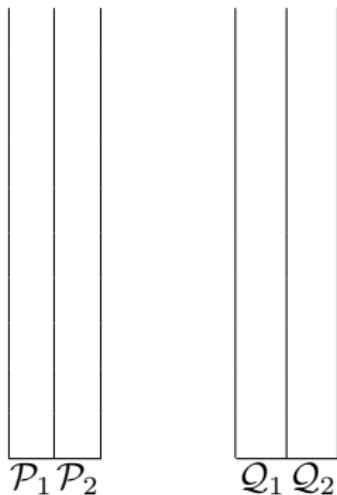
Algorithme KMR

Structures d'index

Extraction de motif

Algorithme KMR [3/3]

$i \quad 1 \ 2 \ 3 \ 4 \ 5 \ 6 \ 7 \ 8 \ 9$
 $s \text{ ROUDOUDOU}$
 $v_4 \text{ - } 1 \ 2 \ - \ 1 \ 2 \ - \ - \ -$



$$\underline{e_8} \Rightarrow \underline{?}$$
$$a = 4, b = 4$$

classe 1 $s[2..5] = s[5..8] = \text{OUDO}$
classe 2 $s[3..6] = s[6..9] = \text{UDOU}$
classe ignorée $s[4..7] = \text{DOUD}$

HLIN608 – Algorithmique du texte.

Alban MANCHERON

Alignement visuel de séquences

Recherche de motif

Extraction de motif

Contexte

Quelques programmes [presque] naïfs

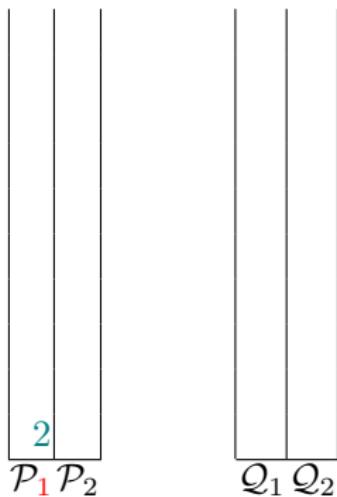
Algorithme KMR

Structures d'index

Extraction de motif

Algorithme KMR [3/3]

$i \quad 1 \ 2 \ 3 \ 4 \ 5 \ 6 \ 7 \ 8 \ 9$
 $s \text{ ROUDOUDOU}$
 $v_4 - \textcolor{red}{1} \ 2 - 1 \ 2 - - -$



$$\underline{e_8} \Rightarrow \underline{?}$$

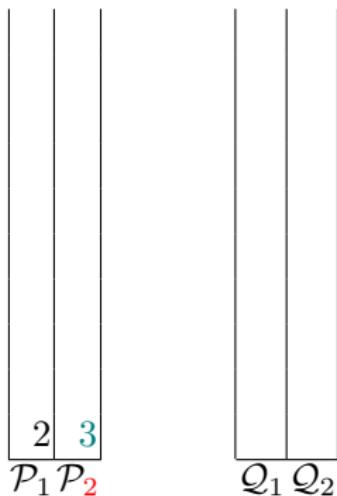
$$a = 4, b = 4$$

classe 1 $s[2..5] = s[5..8] = \text{OUDO}$
classe 2 $s[3..6] = s[6..9] = \text{UDOU}$
classe ignorée $s[4..7] = \text{DOUD}$

Extraction de motif

Algorithme KMR [3/3]

$i \quad 1 \ 2 \ 3 \ 4 \ 5 \ 6 \ 7 \ 8 \ 9$
 $s \text{ ROUDOUDOU}$
 $v_4 - 1 \ 2 - 1 \ 2 - - -$



$$e_8 \Rightarrow ?$$

$a = 4, b = 4$

classe 1 $s[2..5] = s[5..8] = \text{OUDO}$
classe 2 $s[3..6] = s[6..9] = \text{UDOU}$
classe ignorée $s[4..7] = \text{DOUD}$

HLIN608 – Algorithmique du texte.

Alban MANCHERON

Alignement visuel de séquences

Recherche de motif

Extraction de motif

Contexte

Quelques programmes [presque] naïfs

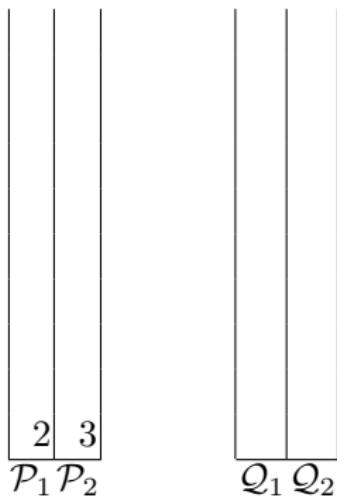
Algorithme KMR

Structures d'index

Extraction de motif

Algorithme KMR [3/3]

$i \quad 1 \ 2 \ 3 \ \textcolor{teal}{4} \ 5 \ 6 \ 7 \ 8 \ 9$
 $s \text{ ROUDOUDOU}$
 $v_4 - 1 \ 2 \ - \ 1 \ 2 \ - \ - \ -$



$$\underline{e_8} \Rightarrow \underline{?}$$
$$a = 4, b = 4$$

classe 1 $s[2..5] = s[5..8] = \text{OUDO}$
classe 2 $s[3..6] = s[6..9] = \text{UDOU}$
classe ignorée $s[4..7] = \text{DOUD}$

HLIN608 – Algorithmique du texte.

Alban MANCHERON

Alignement visuel de séquences

Recherche de motif

Extraction de motif

Contexte

Quelques programmes [presque] naïfs

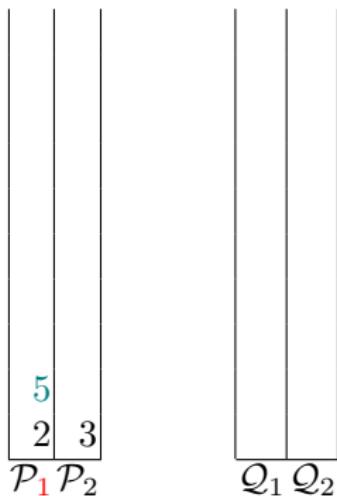
Algorithme KMR

Structures d'index

Extraction de motif

Algorithme KMR [3/3]

<i>i</i>	1	2	3	4	5	6	7	8	9
<i>s</i>	ROUDOUDOU								
<i>v₄</i>	-	1	2	-	1	2	-	-	-



$$\underline{e_8} \Rightarrow \underline{?}$$

$a = 4, b = 4$

classe 1 $s[2..5] = s[5..8] = \text{OUDO}$
classe 2 $s[3..6] = s[6..9] = \text{UDOU}$
classe ignorée $s[4..7] = \text{DOUD}$

HLIN608 – Algorithmique du texte.

Alban MANCHERON

Alignement visuel de séquences

Recherche de motif

Extraction de motif

Contexte

Quelques programmes [presque] naïfs

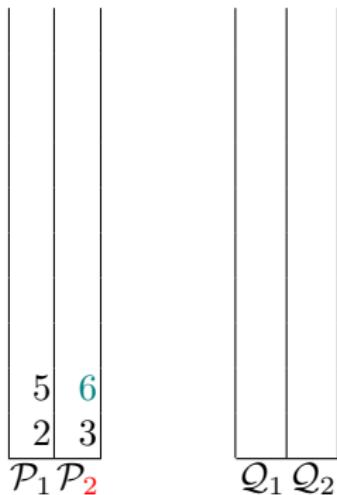
Algorithme KMR

Structures d'index

Extraction de motif

Algorithme KMR [3/3]

i	1	2	3	4	5	6	7	8	9
s	ROUDOUDOU								
v_4	-	1	2	-	1	2	-	-	-



$$\underline{e_8} \Rightarrow \underline{?}$$

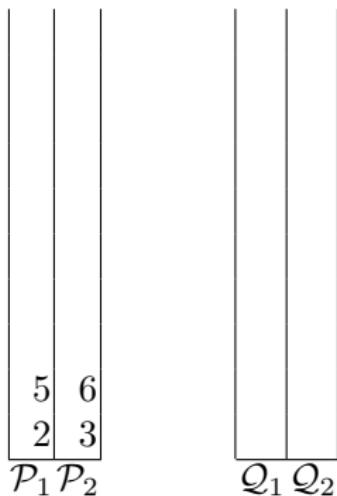
$a = 4, b = 4$

classe 1 $s[2..5] = s[5..8] = \text{OUDO}$
classe 2 $s[3..6] = s[6..9] = \text{UDOU}$
classe ignorée $s[4..7] = \text{DOUD}$

Extraction de motif

Algorithme KMR [3/3]

i	1	2	3	4	5	6	7	8	9
s	ROUDOUDOU								
v_4	-	1	2	-	1	2	-	-	-



$$\underline{e_8} \Rightarrow \underline{?}$$

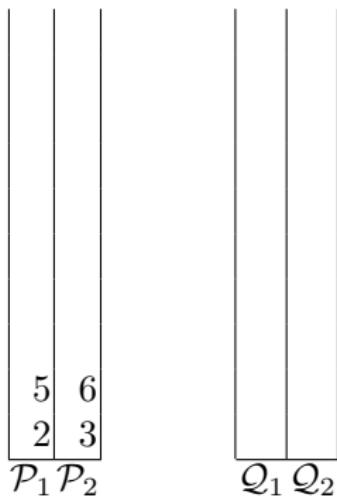
$a = 4, b = 4$

classe 1 $s[2..5] = s[5..8] = \text{OUDO}$
classe 2 $s[3..6] = s[6..9] = \text{UDOU}$
classe ignorée $s[4..7] = \text{DOUD}$

Extraction de motif

Algorithme KMR [3/3]

$i \quad 1 \ 2 \ 3 \ 4 \ 5 \ 6 \ 7 \ 8 \ 9$
 $s \text{ ROUDOUDOU}$
 $v_4 - 1 \ 2 - 1 \ 2 - - -$



$$\underline{e_8} \Rightarrow ?$$

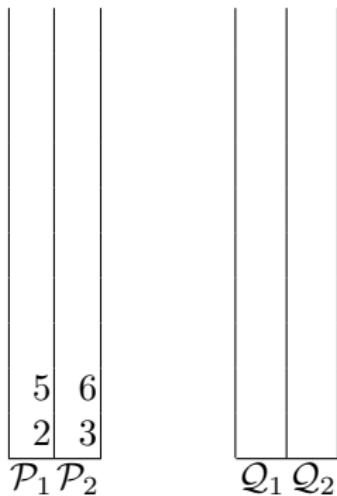
$a = 4, b = 4$

classe 1 $s[2..5] = s[5..8] = \text{OUDO}$
classe 2 $s[3..6] = s[6..9] = \text{UDOU}$
classe ignorée $s[4..7] = \text{DOUD}$

Extraction de motif

Algorithme KMR [3/3]

$i \quad 1 \ 2 \ 3 \ 4 \ 5 \ 6 \ 7 \ 8 \ 9$
 $s \text{ ROUDOUDOU}$
 $v_4 - 1 \ 2 - 1 \ 2 - - -$



$$\underline{e_8} \Rightarrow \underline{?}$$

$a = 4, b = 4$

classe 1 $s[2..5] = s[5..8] = \text{OUDO}$
classe 2 $s[3..6] = s[6..9] = \text{UDOU}$
classe ignorée $s[4..7] = \text{DOUD}$

HLIN608 – Algorithmique du texte.

Alban MANCHERON

Alignement visuel de séquences

Recherche de motif

Extraction de motif

Contexte

Quelques programmes [presque] naïfs

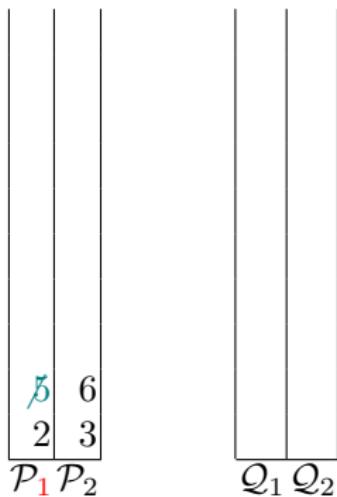
Algorithme KMR

Structures d'index

Extraction de motif

Algorithme KMR [3/3]

$i \quad 1 \ 2 \ 3 \ 4 \ 5 \ 6 \ 7 \ 8 \ 9$
 $s \text{ ROUDOUDOU}$
 $v_4 - 1 \ 2 - 1 \ 2 - - -$



$$\underline{e_8} \Rightarrow ?$$

$a = 4, b = 4$

classe 1 $s[2..5] = s[5..8] = \text{OUDO}$
classe 2 $s[3..6] = s[6..9] = \text{UDOU}$
classe ignorée $s[4..7] = \text{DOUD}$

HLIN608 – Algorithmique du texte.

Alban MANCHERON

Alignement visuel de séquences

Recherche de motif

Extraction de motif

Contexte

Quelques programmes [presque] naïfs

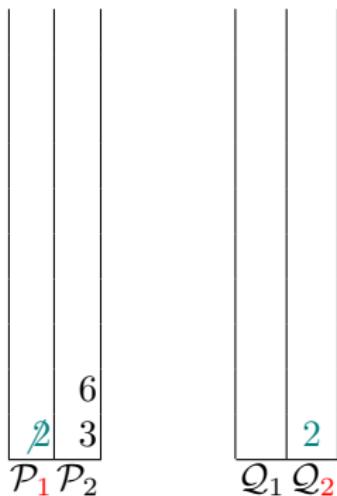
Algorithme KMR

Structures d'index

Extraction de motif

Algorithme KMR [3/3]

$i \quad 1 \ 2 \ 3 \ 4 \ 5 \ \textcolor{teal}{6} \ 7 \ 8 \ 9$
 $s \text{ ROUDOUDOU}$
 $v_4 - 1 \ 2 - 1 \ \textcolor{red}{2} - - -$



$$\underline{e_8} \Rightarrow \underline{?}$$

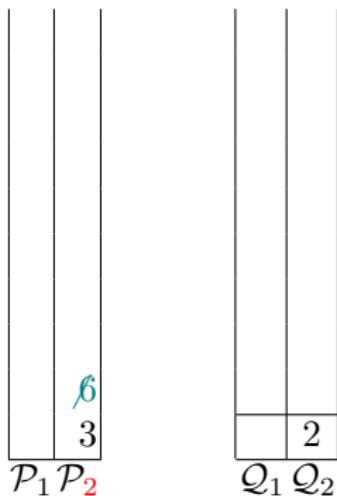
$$a = 4, b = 4$$

classe 1 $s[2..5] = s[5..8] = \text{OUDO}$
classe 2 $s[3..6] = s[6..9] = \text{UDOU}$
classe ignorée $s[4..7] = \text{DOUD}$

Extraction de motif

Algorithme KMR [3/3]

$i \quad 1 \ 2 \ 3 \ 4 \ 5 \ 6 \ 7 \ 8 \ 9$
 $s \text{ ROUDOUDOU}$
 $v_4 - 1 \ 2 - 1 \ 2 - - -$



$$\underline{e_8} \Rightarrow \underline{?}$$

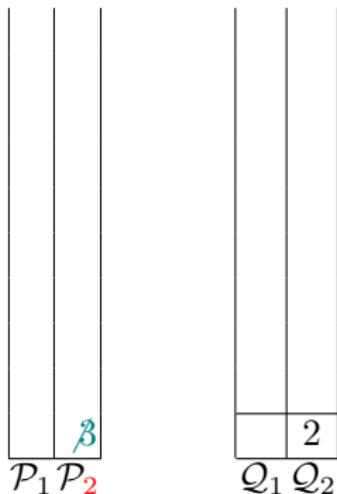
$$a = 4, b = 4$$

classe 1 $s[2..5] = s[5..8] = \text{OUDO}$
classe 2 $s[3..6] = s[6..9] = \text{UDOU}$
classe ignorée $s[4..7] = \text{DOUD}$

Extraction de motif

Algorithme KMR [3/3]

$i \quad 1 \ 2 \ 3 \ 4 \ 5 \ 6 \ 7 \ 8 \ 9$
 $s \text{ ROUDOUDOU}$
 $v_4 - 1 \ 2 - 1 \ 2 - - -$



$$\underline{e_8} \Rightarrow \underline{?}$$

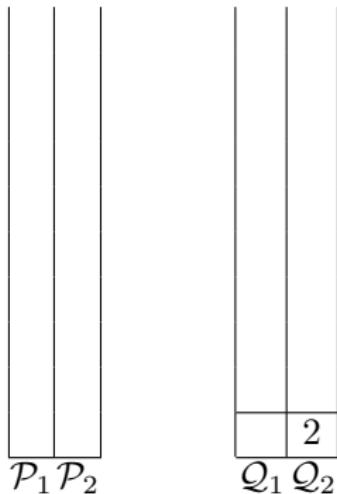
$$a = 4, b = 4$$

classe 1 $s[2..5] = s[5..8] = \text{OUDO}$
classe 2 $s[3..6] = s[6..9] = \text{UDOU}$
classe ignorée $s[4..7] = \text{DOUD}$

Extraction de motif

Algorithme KMR [3/3]

$i \quad 1 \ 2 \ 3 \ 4 \ 5 \ 6 \ 7 \ 8 \ 9$
 $s \text{ ROUDOUDOU}$
 $v_4 - 1 \ 2 - 1 \ 2 - - -$



$$\underline{e_8} \Rightarrow \underline{?}$$

$$a = 4, b = 4$$

classe ignorée $s[2..9] = \text{OUDOUDOU}$

HLIN608 – Algorithmique du texte.

Alban MANCHERON

Alignement visuel de séquences

Recherche de motif

Extraction de motif

Contexte

Quelques programmes [presque] naïfs

Algorithme KMR

Structures d'index

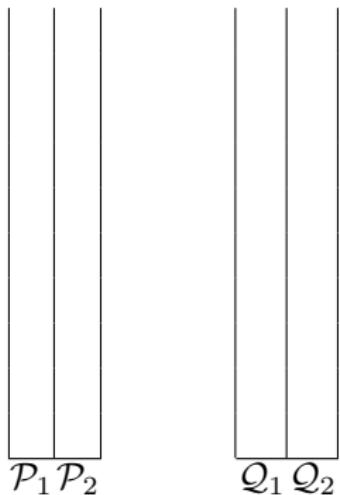
Extraction de motif

Algorithme KMR [3/3]

$i \quad 1 \ 2 \ 3 \ 4 \ 5 \ 6 \ 7 \ 8 \ 9$
 $s \text{ ROUDOUDOU}$
 $v_8 \text{ - - - - - }$

$$\underline{e_8} \Rightarrow 0$$

classe ignorée $s[2..9] = \text{OUDOUDOU}$



HLIN608 – Algorithmique du texte.

Alban MANCHERON

Alignement visuel de séquences

Recherche de motif

Extraction de motif

Contexte

Quelques programmes [presque] naïfs

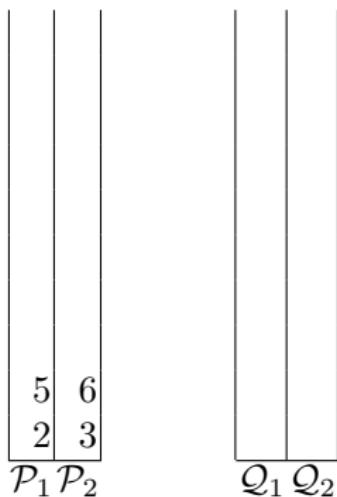
Algorithme KMR

Structures d'index

Extraction de motif

Algorithme KMR [3/3]

i	1	2	3	4	5	6	7	8	9
s	ROUDOUDOU								
v_4	-	1	2	-	1	2	-	-	-



$$\underline{e_6} \Rightarrow \underline{?}$$

$a = 4, b = 2$

classe 1 $s[2..5] = s[5..8] = \text{OUDO}$
classe 2 $s[3..6] = s[6..9] = \text{UDOU}$
classe ignorée $s[4..7] = \text{DOUD}$

HLIN608 – Algorithmique du texte.

Alban MANCHERON

Alignement visuel de séquences

Recherche de motif

Extraction de motif

Contexte

Quelques programmes [presque] naïfs

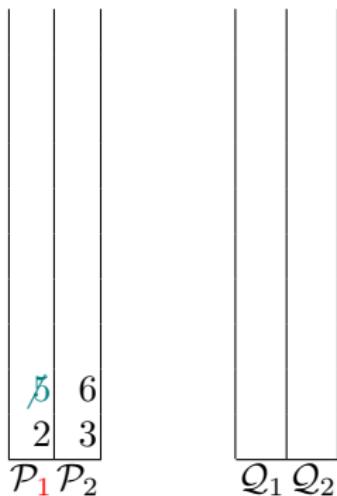
Algorithme KMR

Structures d'index

Extraction de motif

Algorithme KMR [3/3]

$i \quad 1 \ 2 \ 3 \ 4 \ 5 \ 6 \ 7 \ 8 \ 9$
 $s \text{ ROUDOUDOU}$
 $v_4 - 1 \ 2 - 1 \ 2 - - -$



$$\underline{e_6} \Rightarrow \underline{?}$$

$$a = 4, b = 2$$

classe 1 $s[2..5] = s[5..8] = \text{OUDO}$
classe 2 $s[3..6] = s[6..9] = \text{UDOU}$
classe ignorée $s[4..7] = \text{DOUD}$

HLIN608 – Algorithmique du texte.

Alban MANCHERON

Alignement visuel de séquences

Recherche de motif

Extraction de motif

Contexte

Quelques programmes [presque] naïfs

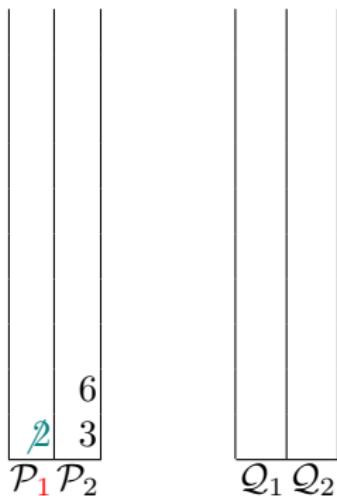
Algorithme KMR

Structures d'index

Extraction de motif

Algorithme KMR [3/3]

$i \quad 1 \ 2 \ 3 \ \textcolor{teal}{4} \ 5 \ 6 \ 7 \ 8 \ 9$
 $s \text{ ROUDOUDOU}$
 $v_4 - 1 \ 2 \ - \ 1 \ 2 \ - \ - \ -$



$$\underline{e_6} \Rightarrow \underline{?}$$

$$a = 4, b = 2$$

classe 1 $s[2..5] = s[5..8] = \text{OUDO}$
classe 2 $s[3..6] = s[6..9] = \text{UDOU}$
classe ignorée $s[4..7] = \text{DOUD}$

HLIN608 – Algorithmique du texte.

Alban MANCHERON

Alignement visuel de séquences

Recherche de motif

Extraction de motif

Contexte

Quelques programmes [presque] naïfs

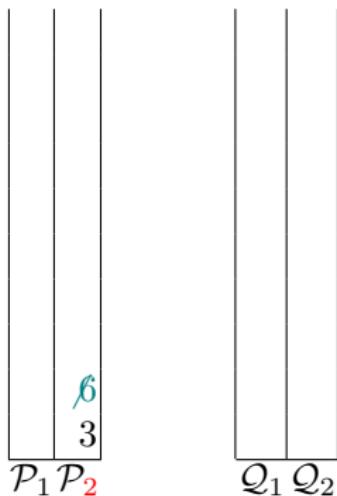
Algorithme KMR

Structures d'index

Extraction de motif

Algorithme KMR [3/3]

$i \quad 1 \ 2 \ 3 \ 4 \ 5 \ 6 \ 7 \ 8 \ 9$
 $s \text{ ROUDOUDOU}$
 $v_4 - 1 \ 2 - 1 \ 2 - - -$



$$\underline{e_6} \Rightarrow \underline{?}$$

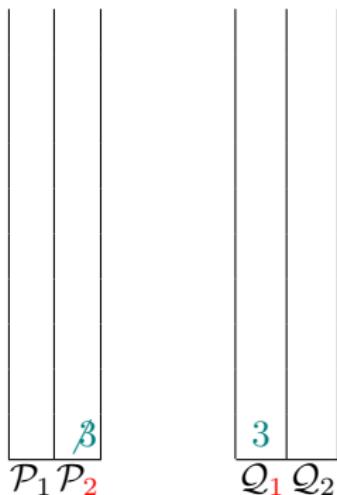
$$a = 4, b = 2$$

classe 1 $s[2..5] = s[5..8] = \text{OUDO}$
classe 2 $s[3..6] = s[6..9] = \text{UDOU}$
classe ignorée $s[4..7] = \text{DOUD}$

Extraction de motif

Algorithme KMR [3/3]

$i \quad 1 \ 2 \ 3 \ 4 \ \textcolor{teal}{5} \ 6 \ 7 \ 8 \ 9$
 $s \text{ ROUDOUDOU}$
 $v_4 - 1 \ 2 - \textcolor{red}{1} \ 2 - - -$



$$\underline{e_6} \Rightarrow \underline{?}$$
$$a = 4, b = 2$$

classe 1 $s[2..5] = s[5..8] = \text{OUDO}$
classe 2 $s[3..6] = s[6..9] = \text{UDOU}$
classe ignorée $s[4..7] = \text{DOUD}$

HLIN608 – Algorithmique du texte.

Alban MANCHERON

Alignement visuel de séquences

Recherche de motif

Extraction de motif

Contexte

Quelques programmes [presque] naïfs

Algorithme KMR

Structures d'index

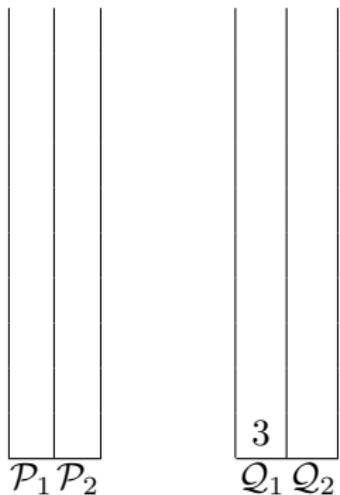
Extraction de motif

Algorithme KMR [3/3]

i	1	2	3	4	5	6	7	8	9
s	ROUDOUDOU								
v_6	-----								

$$\underline{e}_6 \Rightarrow 0$$

classe ignorée $s[3..8] = \text{UDOUDO}$



HLIN608 – Algorithmique du texte.

Alban MANCHERON

Alignement visuel de séquences

Recherche de motif

Extraction de motif

Contexte

Quelques programmes [presque] naïfs

Algorithme KMR

Structures d'index

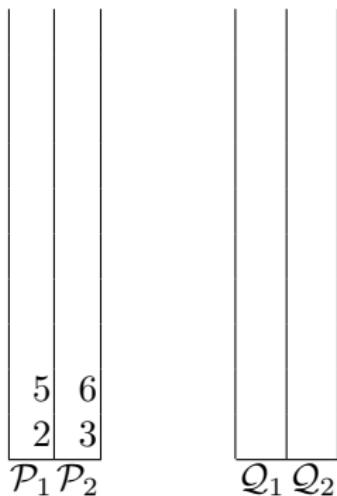
Extraction de motif

Algorithme KMR [3/3]

i	1	2	3	4	5	6	7	8	9
s	ROUDOUDOU								
v_4	-	1	2	-	1	2	-	-	-

$$\underbrace{e_5}_{a=4}, \underbrace{\Rightarrow ?}_{b=1}$$

classe 1 $s[2..5] = s[5..8] = \text{OUDO}$
classe 2 $s[3..6] = s[6..9] = \text{UDOU}$
classe ignorée $s[4..7] = \text{DOUD}$



HLIN608 – Algorithmique du texte.

Alban MANCHERON

Alignement visuel de séquences

Recherche de motif

Extraction de motif

Contexte

Quelques programmes [presque] naïfs

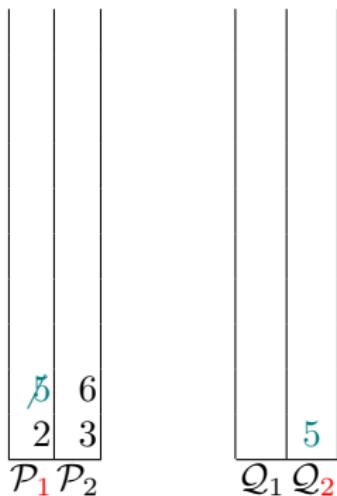
Algorithme KMR

Structures d'index

Extraction de motif

Algorithme KMR [3/3]

i	1	2	3	4	5	6	7	8	9
s	ROUDOUDOU								
v_4	-	1	2	-	1	2	-	-	-



$$\underline{e_5} \Rightarrow \underline{?}$$

$a = 4, b = 1$

classe 1 $s[2..5] = s[5..8] = \text{OUDO}$
classe 2 $s[3..6] = s[6..9] = \text{UDOU}$
classe ignorée $s[4..7] = \text{DOUD}$

Extraction de motif

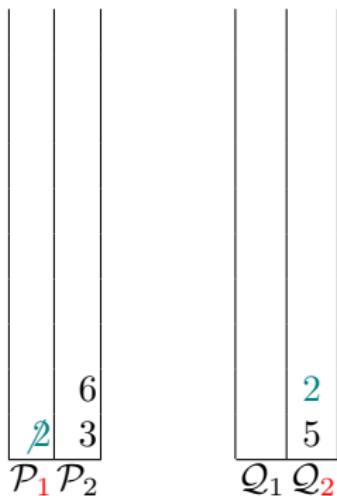
Algorithme KMR [3/3]



<i>i</i>	1	2	3	4	5	6	7	8	9
<i>s</i>	ROUDOUUDOU								
<i>v</i> ₄	-	1	2	-	1	2	-	-	-

$$\begin{array}{l} \text{---} \\ e_5 \Rightarrow ? \\ \text{---} \\ a = 4, b = 1 \end{array}$$

classe 1 $s[2..5] = s[5..8] = \text{OUDO}$
classe 2 $s[3..6] = s[6..9] = \text{UDOU}$
classe ignorée $s[4..7] = \text{DOUD}$



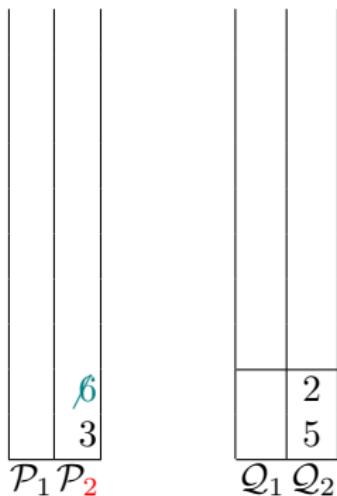
Recherche de motif

Extraction de motif

Extraction de motif

Algorithme KMR [3/3]

$i \quad 1 \ 2 \ 3 \ 4 \ 5 \ 6 \ 7 \ 8 \ 9$
 $s \text{ ROUDOUDOU}$
 $v_4 - 1 \ 2 - 1 \ 2 - - -$



$$\underline{e_5} \Rightarrow \underline{?}$$

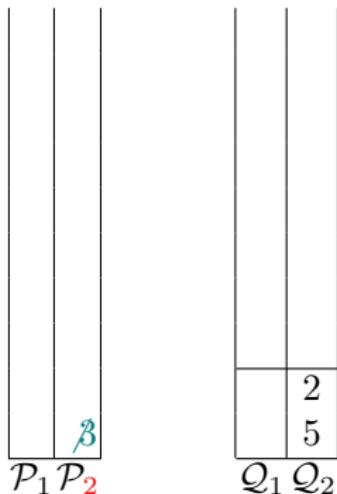
$$a = 4, b = 1$$

classe 1 $s[2..5] = s[5..8] = \text{OUDO}$
classe 2 $s[3..6] = s[6..9] = \text{UDOU}$
classe ignorée $s[4..7] = \text{DOUD}$

Extraction de motif

Algorithme KMR [3/3]

$i \quad 1 \ 2 \ 3 \ \textcolor{teal}{4} \ 5 \ 6 \ 7 \ 8 \ 9$
 $s \text{ ROUDOUDOU}$
 $v_4 - 1 \ 2 \ - \ 1 \ 2 \ - \ - \ -$



$$\underline{e_5} \Rightarrow \underline{?}$$

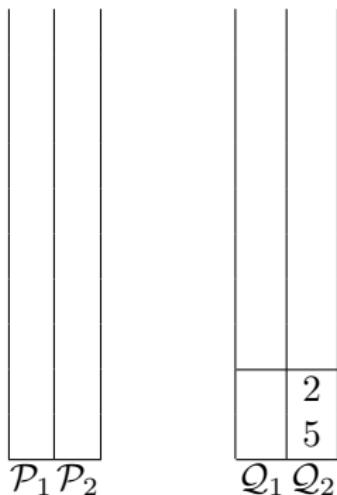
$$a = 4, b = 1$$

classe 1 $s[2..5] = s[5..8] = \text{OUDO}$
classe 2 $s[3..6] = s[6..9] = \text{UDOU}$
classe ignorée $s[4..7] = \text{DOUD}$

Extraction de motif

Algorithme KMR [3/3]

$i \quad 1 \ 2 \ 3 \ 4 \ 5 \ 6 \ 7 \ 8 \ 9$
 $s \text{ ROUDOUDOU}$
 $v_4 - 1 \ 2 - 1 \ 2 - - -$



$$\underline{e_5} \Rightarrow \underline{?}$$

$a = 4, b = 1$

classe 1 $s[2..5] = s[5..8] = \text{OUDO}$
classe 2 $s[3..6] = s[6..9] = \text{UDOU}$
classe ignorée $s[4..7] = \text{DOUD}$

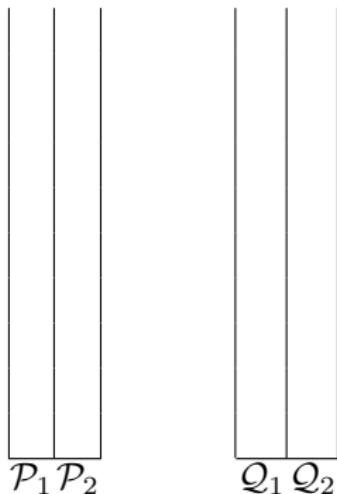
Extraction de motif

Algorithme KMR [3/3]

i	1	2	3	4	5	6	7	8	9
s	R	O	U	D	O	U	D	O	U
v_5	-	1	-	-	1	-	-	-	-

$$\underline{e_5} \Rightarrow 1$$

classe 1 $s[2..5] = s[5..9] = \text{OUDOU}$



HLIN608 – Algorithmique du texte.

Alban MANCHERON

Alignement visuel de séquences

Recherche de motif

Extraction de motif

Contexte

Quelques programmes [presque] naïfs

Algorithme KMR

Structures d'index



Extraction de motif

Algorithme KMR [3/3]

$i \quad 1 \ 2 \ 3 \ 4 \ 5 \ 6 \ 7 \ 8 \ 9$
 $s \text{ ROUDOUDOU}$

$\underline{e_5} \Rightarrow 1$

classe 1 $s[2..5] = s[5..9] = \text{OUDOU}$

That' All, Folks !!!

HLIN608 – Algorithmique du texte.

Alban MANCHERON

Alignement visuel de séquences

Recherche de motif

Extraction de motif

Contexte

Quelques programmes [presque] naïfs

Algorithme KMR

Structures d'index



Extraction de motif

Structures d'index [Hachage]

⇒ Pré-traitement des données en vue de les exploiter plus efficacement

HLIN608 – Algorithmique du texte.

Alban MANCHERON

Alignement visuel de séquences

Recherche de motif

Extraction de motif

Contexte

Quelques programmes [presque] naïfs

Algorithmes KMR

Structures d'index

Hachage

Arbres

Automates

Tables



Extraction de motif

Structures d'index [Hachage]

⇒ Pré-traitement des données en vue de les exploiter plus efficacement

⇒ Un facteur est un préfixe d'un suffixe vecteurs

HLIN608 – Algorithmique du texte.

Alban MANCHERON

Alignement visuel de séquences

Recherche de motif

Extraction de motif

Contexte

Quelques programmes [presque] naïfs

Algorithmes KMR

Structures d'index

Hachage

Arbres

Automates

Tables



Extraction de motif

Structures d'index [Hachage]

⇒ Pré-traitement des données en vue de les exploiter plus efficacement

⇒ Un facteur est un préfixe d'un suffixe vecteurs

Une solution simple

Indexer les suffixes d'une séquence dans une table de hachage.

HLIN608 – Algorithmique du texte.

Alban MANCHERON

Alignement visuel de séquences

Recherche de motif

Extraction de motif

Contexte

Quelques programmes [presque] naïfs

Algorithme KMR

Structures d'index

Hachage

Arbres

Automates

Tables

Extraction de motif

Structures d'index [Hachage]

⇒ Pré-traitement des données en vue de les exploiter plus efficacement

⇒ Un facteur est un préfixe d'un suffixe vecteurs

Une solution simple

Indexer les suffixes d'une séquence dans une table de hachage.

Problèmes liés à la complexité

Espace et temps de construction en $O(n^2)$.

HLIN608 – Algorithmique du texte.

Alban MANCHERON

Alignement visuel de séquences

Recherche de motif

Extraction de motif

Contexte

Quelques programmes [presque] naïfs

Algorithmes KMR

Structures d'index

Hachage

Arbres

Automates

Tables

Extraction de motif

Structures d'index [Arbres]



Définition (Arbre [enraciné])

≡ graphe non orienté connexe et sans cycle.

- Un **nœud** est un sommet de l'arbre
- La **racine** est un nœud particulier de l'arbre qui lui confère une orientation
 - ⇒ La **racine** est le seul nœud sans prédécesseur
- Une **feuille** est un nœud de l'arbre sans successeurs
- Un nœud est dit **interne** s'il n'est ni la racine ni une feuille
- La **profondeur** d'un nœud est la longueur du chemin (unique) de la racine vers ce nœud.
- Le **niveau** d'un nœud est sa profondeur plus 1.
- La **hauteur** est égale à la plus grande profondeur de l'arbre
- La **largeur** est égale au plus grand nombre de nœud au même niveau de l'arbre

HLIN608 – Algorithmique du texte.

Alban MANCHERON

Alignement visuel de séquences

Recherche de motif

Extraction de motif

Contexte

Quelques programmes [presque] naïfs

Algorithmme KMR

Structures d'index

Hachage

Arbres

Automates

Tables

Extraction de motif

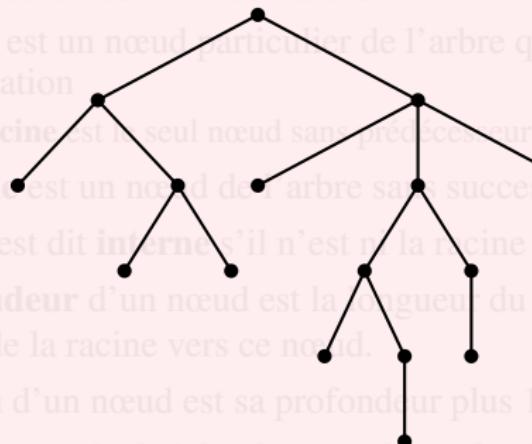
Structures d'index [Arbres]



Définition (Arbre [enraciné])

graphie non orientée connexe et sans cycle.

- Un **nœud** est un sommet de l'arbre
- La **racine** est un nœud particulier de l'arbre qui lui confère une orientation
 - ⇒ La racine est le seul nœud sans prédecesseur
- Une **feuille** est un nœud de l'arbre sans successeurs
- Un nœud est dit **interne** s'il n'est ni la racine ni une feuille
- La **profondeur** d'un nœud est la longueur du chemin (unique) de la racine vers ce nœud.
- Le **niveau** d'un nœud est sa profondeur plus 1.
- La **hauteur** est égale à la plus grande profondeur de l'arbre
- La **largeur** est égale au plus grand nombre de nœud au même niveau de l'arbre



Extraction de motif

Structures d'index [Arbres]

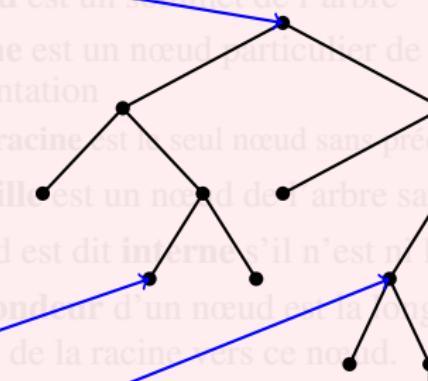


Définition (Arbre [enraciné])

graphie non orientée connexe et sans cycle.

racine

- Un nœud est un sommet de l'arbre
- La racine est un nœud particulier de l'arbre qui lui confère une orientation
 ⇒ La racine est le seul nœud sans prédecesseur

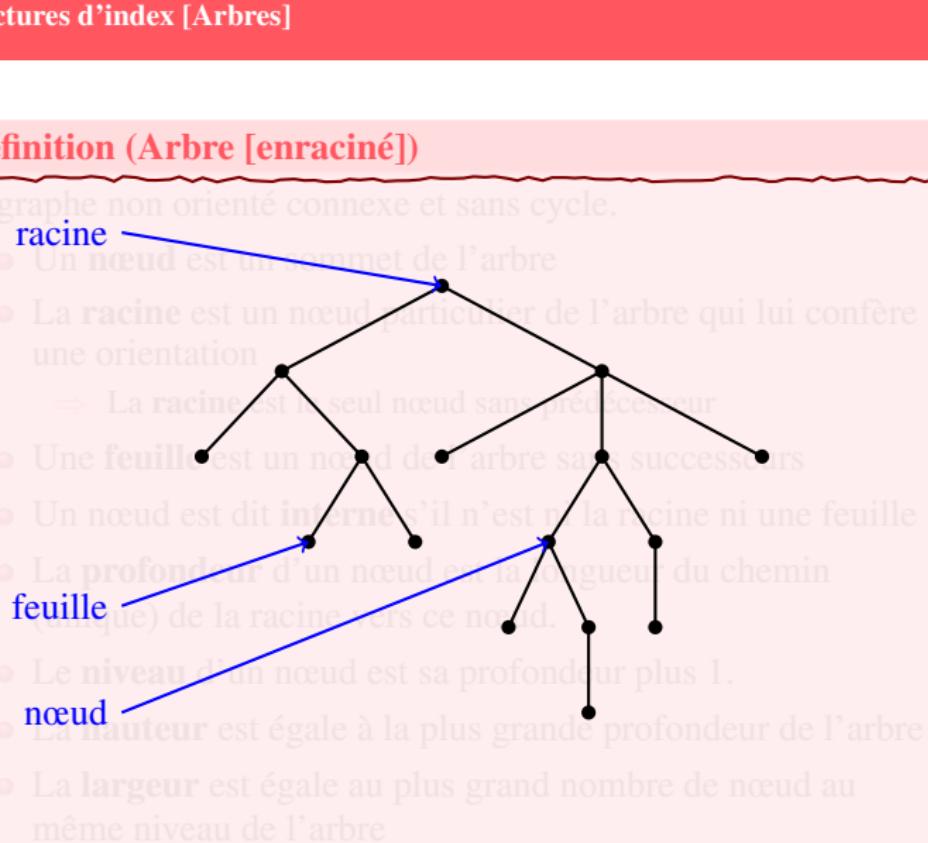


- Une feuille est un nœud de l'arbre sans successeurs

- Un nœud est dit interne s'il n'est ni la racine ni une feuille
- La profondeur d'un nœud est la longueur du chemin (unique) de la racine vers ce nœud.

feuille

- Le niveau d'un nœud est sa profondeur plus 1.
- La hauteur est égale à la plus grande profondeur de l'arbre
- La largeur est égale au plus grand nombre de nœud au même niveau de l'arbre



HLIN608 – Algorithmique du texte.

Alban MANCHERON

Alignement visuel de séquences

Recherche de motif

Extraction de motif

Contexte

Quelques programmes [presque] naïfs

Algorithmes KMR

Structures d'index

Hachage

Arbres

Automates

Tables

Extraction de motif

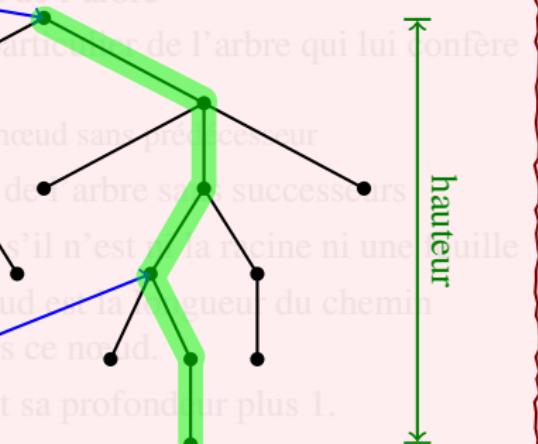
Structures d'index [Arbres]

Définition (Arbre [enraciné])

graphie non orientée connexe et sans cycle.

racine

- Un nœud est un sommet de l'arbre
- La racine est un nœud particulier de l'arbre qui lui confère une orientation
→ La racine est le seul nœud sans prédecesseur



feuille

- Une feuille est un nœud de l'arbre sans successeurs
- Un nœud est dit interne s'il n'est ni la racine ni une feuille
- La profondeur d'un nœud est la longueur du chemin (unique) de la racine vers ce nœud.
- Le niveau d'un nœud est sa profondeur plus 1.
- La hauteur est égale à la plus grande profondeur de l'arbre
- La largeur est égale au plus grand nombre de nœud au même niveau de l'arbre

nœud

Extraction de motif

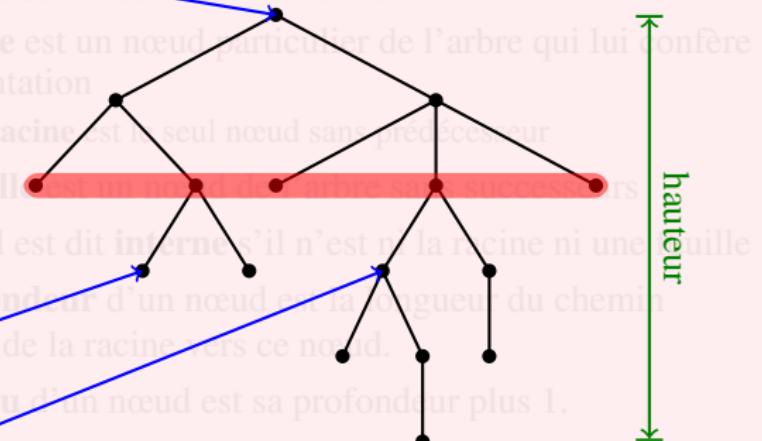
Structures d'index [Arbres]

Définition (Arbre [enraciné])

graphie non orientée connexe et sans cycle.

racine

- Un nœud est un sommet de l'arbre
- La racine est un nœud particulier de l'arbre qui lui confère une orientation
→ La racine est le seul nœud sans prédecesseur



feuille

- Un nœud est dit interne s'il n'est ni la racine ni une feuille

nœud

- Le niveau d'un nœud est sa profondeur plus 1.
- La hauteur est égale à la plus grande profondeur de l'arbre
- La largeur est égale au plus grand nombre de nœuds au même niveau de l'arbre

Extraction de motif

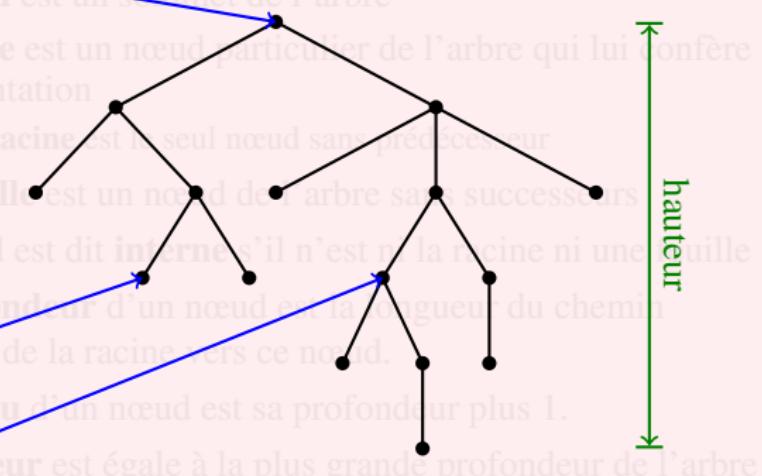
Structures d'index [Arbres]

Définition (Arbre [enraciné])

graphie non orientée connexe et sans cycle.

racine

- Un nœud est un sommet de l'arbre
- La racine est un nœud particulier de l'arbre qui lui confère une orientation
→ La racine est le seul nœud sans prédecesseur



feuille

- Une feuille est un nœud de l'arbre sans successeurs
- Un nœud est dit interne s'il n'est ni la racine ni une feuille

nœud

- Le niveau d'un nœud est sa profondeur plus 1.
- La hauteur est égale à la plus grande profondeur de l'arbre
- La largeur est égale au plus grand nombre de nœuds au même niveau de l'arbre

Extraction de motif

Structures d'index [Arbres]

Définition (**Trie**)

≡ Arbre enraciné construit à partir d'un ensemble de mots (structure proposée en 1960 par Edward FREDKIN).

- Chaque chemin de la racine à une feuille de l'arbre étiquette un des mots de l'ensemble
 - Tous les mots de l'ensemble sont représentés
 - Aucun mot supplémentaire n'y est présent
- ⇒ Tous les mots de l'ensemble représenté qui partagent le même préfixe sont situés dans le sous-arbre issu du nœud associé à ce préfixe
- Le terme *Trie* provient du mot anglais retrieval



Extraction de motif

Structures d'index [Arbres]

Définition (*Trie*)

≡ Arbre enraciné construit à partir d'un ensemble de mots
(structure proposée en 1960 par Edward FREDKIN).

- Chaque chemin de la racine à une feuille de l'arbre étiquette un des mots de l'ensemble
 - Tous les mots de l'ensemble sont représentés
 - Aucun mot supplémentaire n'y est présent
- ⇒ Tous les mots de l'ensemble représenté qui partagent le même préfixe sont situés dans le sous-arbre issu du nœud associé à ce préfixe
- Le terme *Trie* provient du mot anglais retrieval

Ajout d'un symbole marquant la fin du mot ⇒ cas où un mot et son préfixe sont dans l'ensemble à indexer...

HLIN608 – Algorithmique du texte.

Alban MANCHERON

Alignement visuel de séquences

Recherche de motif

Extraction de motif

Contexte

Quelques programmes [presque] naïfs

Algorithme KMR

Structures d'index

Hachage

Arbres

Automates

Tables



Extraction de motif

Structures d'index [Arbres]

Définition (*Trie*)

- Arbre racine construit à partir d'un ensemble de mots
- structure proposée en 1937 par Fredkin
- Chaque nœud du tableau fait partie d'un arbre étiqueté avec un des mots de l'ensemble
 - Tous les mots de l'ensemble sont représentés
 - Aucun mot supplémentaire n'y est présent
- Tous les mots de l'ensemble représenté qui partagent le même préfixe sont situés dans le sous-arbre issu du nœud associé à ce préfixe
- Le terme *Trie* provient du mot anglais *retrieval*

Ajout d'un symbole marquant la fin du mot \Rightarrow cas où un mot et son préfixe sont dans l'ensemble à indexer...

HLIN608 – Algorithmique du texte.

Alban MANCHERON

Alignement visuel de séquences

Recherche de motif

Extraction de motif

Contexte

Quelques programmes [presque] naïfs

Algorithmes KMR

Structures d'index

Hachage

Arbres

Automates

Tables

Extraction de motif

Structures d'index [Arbres]

Définition (Trie)

Trie de l'ensemble:

{DOUX, DO, ROUX, ROUDOUDOU}

- Chaque élément de l'ensemble est étiqueté par un des mots de l'ensemble

- Tous les mots de l'ensemble sont représentés
- Aucun mot supplémentaire n'y est présent

⇒ Tous les mots de l'ensemble représenté qui partagent le même préfixe sont situés dans le sous-arbre issu du nœud associé à ce préfixe

- Le terme *Trie* provient du mot anglais *retrieval*

Ajout d'un symbole marquant la fin du mot ⇒ cas où un mot et son préfixe sont dans l'ensemble à indexer...

Extraction de motif

Structures d'index [Arbres]

Définition (Trie)

Trie de l'ensemble:

{DOUX, DO, ROUX, ROUDOUDOU}

- Chaque mot de l'ensemble est représenté par un nœud de l'arbre étiqueté avec un des mots de l'ensemble

- Tous les mots de l'ensemble sont représentés
- Aucun mot supérieur à l'ensemble n'y est présent

⇒ Tous les mots de l'ensemble représenté qui partagent le même préfixe sont situés dans le sous-arbre issu du nœud associé à ce préfixe

- Le terme *Trie* provient du mot anglais *retrieval*

Ajout d'un symbole marquant la fin du mot ⇒ cas où un mot et son préfixe sont dans l'ensemble à indexer...

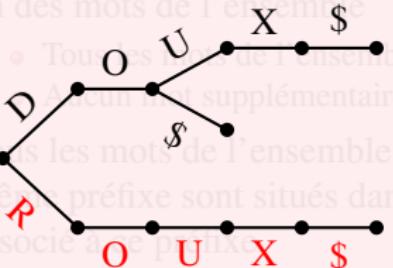
Extraction de motif

Structures d'index [Arbres]

Définition (Trie)

Trie de l'ensemble:
{DOUX, DO, ROUX, ROUDOUDOU}

- Chaque nœud de l'arbre est étiqueté par un des mots de l'ensemble
 - Toutes les mots de l'ensemble sont représentés
 - Aucun mot supplémentaire n'y est présent
- ⇒ Tous les mots de l'ensemble représenté qui partagent le même préfixe sont situés dans le sous-arbre issu du nœud associé à ce préfixe
- Le terme *Trie* provient du mot anglais *retrieval*



Ajout d'un symbole marquant la fin du mot ⇒ cas où un mot et son préfixe sont dans l'ensemble à indexer...

Extraction de motif

Structures d'index [Arbres]

Définition (Trie)

Trie de l'ensemble:
{DOUX, DO, ROUX, ROUDOUDOU}

- Chaque noeud de l'arbre est une feuille. L'arbre étiqueté un des mots de l'ensemble
 - Toutes les mots de l'ensemble sont représentés
 - Aucun mot supplémentaire n'y est présent
- ⇒ Tous les mots de l'ensemble présenté qui partagent le même préfixe sont stockés dans le sous-arbre issu du noeud associé à ce préfixe.
- Le terme *Trie* provient du mot anglais *retrieval*



Ajout d'un symbole marquant la fin du mot ⇒ cas où un mot et son préfixe sont dans l'ensemble à indexer...

Extraction de motif

Structures d'index [Arbres]



Trie d'un ensemble de n mots

- Nombre de feuilles ?
- Hauteur ?

HLIN608 – Algorithmique du texte.

Alban MANCHERON

Alignement visuel de séquences

Recherche de motif

Extraction de motif

Contexte

Quelques programmes [presque] naïfs

Algorithmme KMR

Structures d'index

Hachage

Arbres

Automates

Tables



Extraction de motif

Structures d'index [Arbres]

Trie d'un ensemble de n mots

- Nombre de feuilles ?

 ⇒ n feuilles

- Hauteur ?

HLIN608 – Algorithmique du texte.

Alban MANCHERON

Alignement visuel de séquences

Recherche de motif

Extraction de motif

Contexte

Quelques programmes [presque] naïfs

Algorithmme KMR

Structures d'index

Hachage

Arbres

Automates

Tables



Extraction de motif

Structures d'index [Arbres]

Trie d'un ensemble de n mots

- Nombre de feuilles ?

⇒ n feuilles

- Hauteur ?

⇒ longueur du plus grand mot de l'ensemble

HLIN608 – Algorithmique du texte.

Alban MANCHERON

Alignement visuel de séquences

Recherche de motif

Extraction de motif

Contexte

Quelques programmes [presque] naïfs

Algorithmme KMR

Structures d'index

Hachage

Arbres

Automates

Tables



Extraction de motif

Structures d'index [Arbres]

Définition (*Patricia Trie*)

≡ version compacte d'un *Trie* (structure proposée en 1968 par Donald MORRISON).

- seuls la racine, les nœuds internes ayant au moins deux fils, ainsi que les feuilles sont conservés
- = *Practical Algorithm To Retrieve Information Coded In Alphanumeric Trie*

HLIN608 – Algorithmique du texte.

Alban MANCHERON

Alignement visuel de séquences

Recherche de motif

Extraction de motif

Contexte

Quelques programmes [presque] naïfs

Algorithmme KMR

Structures d'index

Hachage

Arbres

Automates

Tables



Extraction de motif

Structures d'index [Arbres]

Définition (*Patricia Trie*)

= version compacte d'un Trie (structure proposée en 1968 par

Donald MORRISON)

**Patricia Trie de l'ensemble:
{DOUX, DO, ROUX, ROUDOUDOU}**

- seuls la racine, les nœuds internes ayant au moins deux fils,
ainsi que les feuilles sont conservés

= *Practical Algorithm To Retrieve Information Coded In
Alphanumeric Trie*

•

HLIN608 – Algorithmique du
texte.

Alban MANCHERON

Alignement visuel de séquences

Recherche de motif

Extraction de motif

Contexte

Quelques programmes [presque] naïfs

Algorithme KMR

Structures d'index

Hachage

Arbres

Automates

Tables

Extraction de motif

Structures d'index [Arbres]

Définition (*Patricia Trie*)

= version compacte d'un Trie (structure proposée en 1968 par

Donald MORRISON)

**Patricia Trie de l'ensemble:
{DOUX, DO, ROUX, ROUDOUDOU}**

- seuls la racine, les nœuds internes ayant au moins deux fils,
ainsi que les feuilles sont conservés

= *Practical Algorithm To Retrieve Information Coded In
Alphanumeric Trie*

DOUX\$



HLIN608 – Algorithmique du
texte.

Alban MANCHERON

Alignement visuel de séquences

Recherche de motif

Extraction de motif

Contexte

Quelques programmes [presque] naïfs

Algorithme KMR

Structures d'index

Hachage

Arbres

Automates

Tables

Extraction de motif

Structures d'index [Arbres]

Définition (*Patricia Trie*)

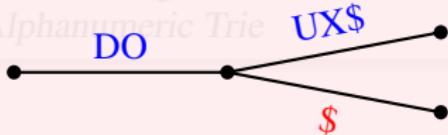
= version compacte d'un Trie (structure proposée en 1968 par

Donald MORRISON)

Patricia Trie de l'ensemble:
{DOUX, DO, ROUX, ROUDOUDOU}

- seuls la racine, les nœuds internes ayant au moins deux fils, ainsi que les feuilles sont conservés

= *Practical Algorithm To Retrieve Information Coded In Alphanumeric Trie*



HLIN608 – Algorithmique du texte.

Alban MANCHERON

Alignement visuel de séquences

Recherche de motif

Extraction de motif

Contexte

Quelques programmes [presque] naïfs

Algorithme KMR

Structures d'index

Hachage

Arbres

Automates

Tables

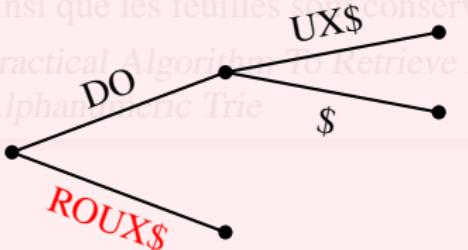
Extraction de motif

Structures d'index [Arbres]

Définition (*Patricia Trie*)

*Patricia Trie de l'ensemble:
{DOUX, DO, ROUX, ROUDOUDOU}*

- seuls la racine, les noeuds internes ayant au moins deux fils, ainsi que les feuilles sont conservés



HLIN608 – Algorithmique du texte.

Alban MANCHERON

Alignement visuel de séquences

Recherche de motif

Extraction de motif

Contexte

Quelques programmes [presque] naïfs

Algorithmme KMR

Structures d'index

Hachage

Arbres

Automates

Tables

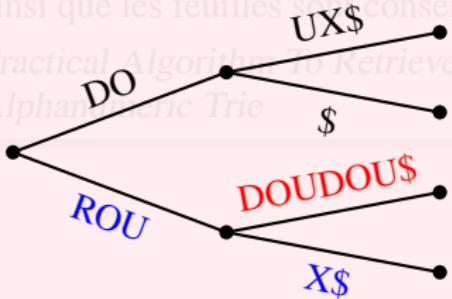
Extraction de motif

Structures d'index [Arbres]

Définition (Patricia Trie)

**Patricia Trie de l'ensemble:
 {DOUX, DO, ROUX, ROUDOUDOU}**

- seuls la racine, les noeuds internes ayant au moins deux fils, ainsi que les feuilles sont conservés



HLIN608 – Algorithmique du texte.

Alban MANCHERON

Alignement visuel de séquences

Recherche de motif

Extraction de motif

Contexte

Quelques programmes [presque] naïfs

Algorithmme KMR

Structures d'index

Hachage

Arbres

Automates

Tables



Extraction de motif

Structures d'index [Arbres]

Définition (Arbre [compact] des suffixes)

≡ [Patricia] Trie des suffixes d'un mot

- construction linéaire (temps/espace)
 - ⇒ Peter WEINER en 1973
- construction linéaire *on-line*
 - ⇒ Esko UKKONEN en 1995

HLIN608 – Algorithmique du texte.

Alban MANCHERON

Alignement visuel de séquences

Recherche de motif

Extraction de motif

Contexte

Quelques programmes [presque] naïfs

Algorithme KMR

Structures d'index

Hachage

Arbres

Automates

Tables

Extraction de motif

Structures d'index [Arbres]

Définition (Arbre [compact] des suffixes)

[Patricia] Trie des suffixes d'un mot

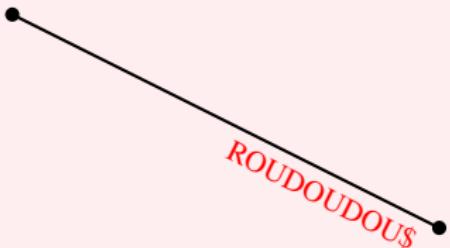
Arbre [compact] des suffixes du mot **ROUDOUDOU**

- construction linéaire

⇒ Peter WEINER en 1973

- construction linéaire *on-line*

⇒ Esko UKKONEN en 1995



Extraction de motif

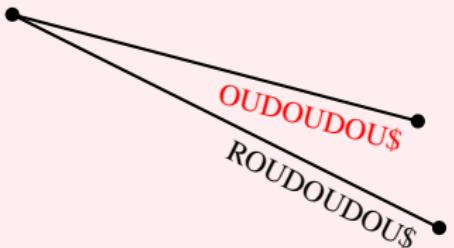
Structures d'index [Arbres]

Définition (Arbre [compact] des suffixes)

[Patricia] Trie des suffixes d'un mot

Arbre [compact] des suffixes du mot **ROUDOUDOU**

- construction linéaire
⇒ Peter WEINER en 1973
- construction linéaire *on-line*
⇒ Esko UKKONEN en 1995



Extraction de motif

Structures d'index [Arbres]

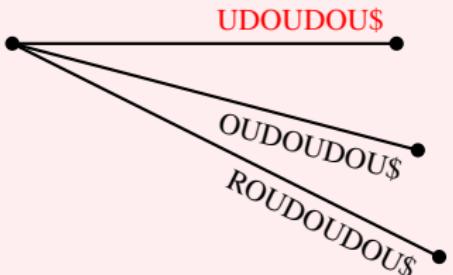
Définition (Arbre [compact] des suffixes)

[Patricia] Trie des suffixes d'un mot

• construction linéaire
ROUDOUDOU

⇒ Peter WEINER en 1973

• construction linéaire *on-line*
⇒ Esko UKKONEN en 1995



HLIN608 – Algorithmique du texte.

Alban MANCHERON

Alignement visuel de séquences

Recherche de motif

Extraction de motif

Contexte

Quelques programmes [presque] naïfs

Algorithme KMR

Structures d'index

Hachage

Arbres

Automates

Tables

Extraction de motif

Structures d'index [Arbres]

Définition (Arbre [compact] des suffixes)

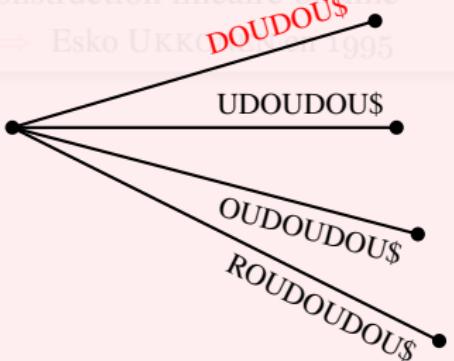
[Patricia] Trie des suffixes d'un mot

• construction linéaire
Arbre [compact] des suffixes du mot
ROUDOUDOU****

⇒ Peter WEINER en 1973

• construction linéaire *on-line*

⇒ Esko UKKonen - 1995



Extraction de motif

Structures d'index [Arbres]



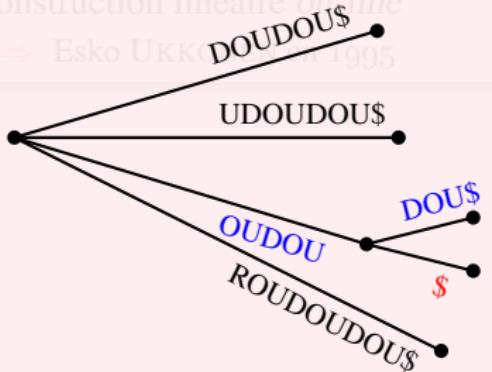
Définition (Arbre [compact] des suffixes)

[Patricia] Trie des suffixes d'un mot

• construction linéaire
Arbre [compact] des suffixes du mot
ROUDOUUDOU

⇒ Peter WEINER en 1973

• construction linéaire *on-line*
⇒ Esko UKKonen - 1995



HLIN608 – Algorithmique du texte.

Alban MANCHERON

Alignement visuel de séquences

Recherche de motif

Extraction de motif

Contexte

Quelques programmes [presque] naïfs

Algorithme KMR

Structures d'index

Hachage

Arbres

Automates

Tables

Extraction de motif

Structures d'index [Arbres]

Définition (Arbre [compact] des suffixes)

[Patricia] Trie des suffixes d'un mot

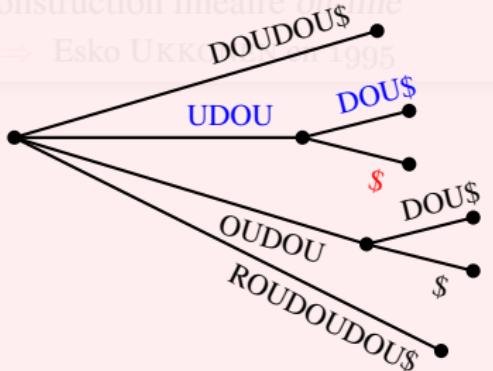
Arbre [compact] des suffixes du mot ROUDOUDOU

- construction linéaire

⇒ Peter WEINER en 1973

- construction linéaire *on-line*

⇒ Esko UKKONEN - 1995



Extraction de motif

Structures d'index [Arbres]

Définition (Arbre [compact] des suffixes)

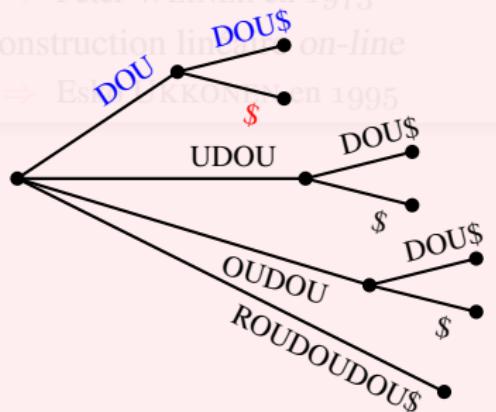
[Patricia] Trie des suffixes d'un mot

• construction linéaire
Arbre [compact] des suffixes du mot
ROUDOU^{DOU}

⇒ Peter WEINER en 1973

• construction linéaire *on-line*

⇒ Esko KORKEALA en 1995



HLIN608 – Algorithmique du texte.

Alban MANCHERON

Alignement visuel de séquences

Recherche de motif

Extraction de motif

Contexte

Quelques programmes [presque] naïfs

Algorithme KMR

Structures d'index

Hachage

Arbres

Automates

Tables

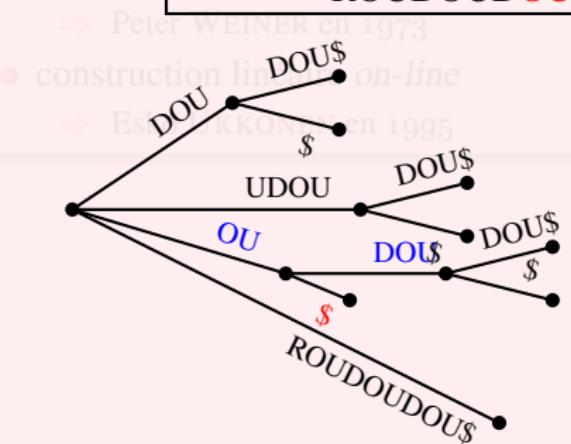
Extraction de motif

Structures d'index [Arbres]

Définition (Arbre [compact] des suffixes)

[Patricia] Trie des suffixes d'un mot

Arbre [compact] des suffixes du mot ROUDOUDOU



Extraction de motif

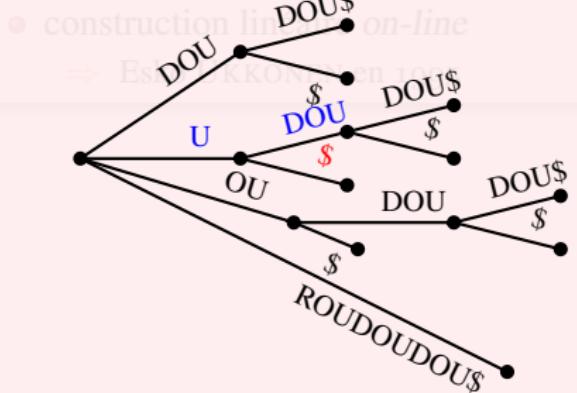
Structures d'index [Arbres]

Définition (Arbre [compact] des suffixes)

[Patricia] Trie des suffixes d'un mot

Arbre [compact] des suffixes du mot ROUDOUDOU

⇒ Peter WEINER en 1973



- construction linéaire *on-line*

⇒ Esko KORKEALA en 1973

HLIN608 – Algorithmique du texte.

Alban MANCHERON

Alignement visuel de séquences

Recherche de motif

Extraction de motif

Contexte

Quelques programmes [presque] naïfs

Algorithmme KMR

Structures d'index

Hachage

Arbres

Automates

Tables

Extraction de motif

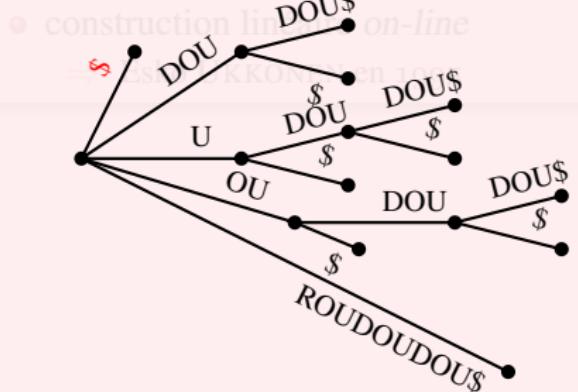
Structures d'index [Arbres]

Définition (Arbre [compact] des suffixes)

[Patricia] Trie des suffixes d'un mot

Arbre [compact] des suffixes du mot ROUDOUDOU

⇒ Peter WEINER en 1973



Extraction de motif

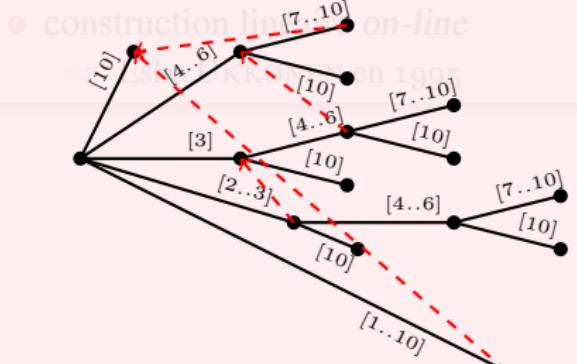
Structures d'index [Arbres]

Définition (Arbre [compact] des suffixes)

[Patricia] Trie des suffixes d'un mot

Arbre [compact] des suffixes du mot ROUDOUDOU

⇒ Peter WEINER en 1973



Extraction de motif

Structures d'index [Automates]



Définition (Automate à états finis)

\equiv quintuplet $(\mathcal{Q}, \mathcal{I}, \mathcal{F}, \Sigma, \delta)$

HLIN608 – Algorithmique du texte.

Alban MANCHERON

Alignement visuel de séquences

Recherche de motif

Extraction de motif

Contexte

Quelques programmes [presque] naïfs

Algorithme KMR

Structures d'index

Hachage

Arbres

Automates

Tables

Extraction de motif

Structures d'index [Automates]



Définition (Automate à états finis)

\equiv quintuplet $(\mathcal{Q}, \mathcal{I}, \mathcal{F}, \Sigma, \delta)$

- \mathcal{Q} : ensemble fini d'états

HLIN608 – Algorithmique du texte.

Alban MANCHERON

Alignement visuel de séquences

Recherche de motif

Extraction de motif

Contexte

Quelques programmes [presque] naïfs

Algorithmes KMR

Structures d'index

Hachage

Arbres

Automates

Tables



Extraction de motif

Structures d'index [Automates]

Définition (Automate à états finis)

≡ quintuplet $(\mathcal{Q}, \mathcal{I}, \mathcal{F}, \Sigma, \delta)$

- \mathcal{Q} : ensemble fini d'états
- $\mathcal{I}(\subseteq Q)$: ensemble des états initiaux

HLIN608 – Algorithmique du texte.

Alban MANCHERON

Alignement visuel de séquences

Recherche de motif

Extraction de motif

Contexte

Quelques programmes [presque] naïfs

Algorithme KMR

Structures d'index

Hachage

Arbres

Automates

Tables



Extraction de motif

Structures d'index [Automates]

Définition (Automate à états finis)

≡ quintuplet $(\mathcal{Q}, \mathcal{I}, \mathcal{F}, \Sigma, \delta)$

- \mathcal{Q} : ensemble fini d'états
- $\mathcal{I}(\subseteq Q)$: ensemble des états initiaux
- $\mathcal{F}(\subseteq Q)$: ensemble des états finaux

HLIN608 – Algorithmique du texte.

Alban MANCHERON

Alignement visuel de séquences

Recherche de motif

Extraction de motif

Contexte

Quelques programmes [presque] naïfs

Algorithme KMR

Structures d'index

Hachage

Arbres

Automates

Tables

Extraction de motif

Structures d'index [Automates]



Définition (Automate à états finis)

≡ quintuplet $(\mathcal{Q}, \mathcal{I}, \mathcal{F}, \Sigma, \delta)$

- \mathcal{Q} : ensemble fini d'états
- $\mathcal{I}(\subseteq Q)$: ensemble des états initiaux
- $\mathcal{F}(\subseteq Q)$: ensemble des états finaux
- Σ : alphabet

HLIN608 – Algorithmique du texte.

Alban MANCHERON

Alignement visuel de séquences

Recherche de motif

Extraction de motif

Contexte

Quelques programmes [presque] naïfs

Algorithme KMR

Structures d'index

Hachage

Arbres

Automates

Tables



Extraction de motif

Structures d'index [Automates]

Définition (Automate à états finis)

≡ quintuplet $(\mathcal{Q}, \mathcal{I}, \mathcal{F}, \Sigma, \delta)$

- \mathcal{Q} : ensemble fini d'états
- $\mathcal{I}(\subseteq \mathcal{Q})$: ensemble des états initiaux
- $\mathcal{F}(\subseteq \mathcal{Q})$: ensemble des états finaux
- Σ : alphabet
- $\delta \in \mathcal{P}(\mathcal{Q} \times \Sigma \times \mathcal{Q})$: relation de transition

HLIN608 – Algorithmique du texte.

Alban MANCHERON

Alignement visuel de séquences

Recherche de motif

Extraction de motif

Contexte

Quelques programmes [presque] naïfs

Algorithmme KMR

Structures d'index

Hachage

Arbres

Automates

Tables



Extraction de motif

Structures d'index [Automates]

Définition (Automate à états finis)

≡ quintuplet $(\mathcal{Q}, \mathcal{I}, \mathcal{F}, \Sigma, \delta)$

- \mathcal{Q} : ensemble fini d'états
- $\mathcal{I}(\subseteq \mathcal{Q})$: ensemble des états initiaux
- $\mathcal{F}(\subseteq \mathcal{Q})$: ensemble des états finaux
- Σ : alphabet
- $\delta \in \mathcal{P}(\mathcal{Q} \times \Sigma \times \mathcal{Q})$: relation de transition

Propriétés sympathiques (ou pas !)

⇒ Déterminisme

HLIN608 – Algorithmique du texte.

Alban MANCHERON

Alignement visuel de séquences

Recherche de motif

Extraction de motif

Contexte

Quelques programmes [presque] naïfs

Algorithme KMR

Structures d'index

Hachage

Arbres

Automates

Tables

Extraction de motif

Structures d'index [Automates]



Définition (Automate à états finis)

≡ quintuplet $(\mathcal{Q}, \mathcal{I}, \mathcal{F}, \Sigma, \delta)$

- \mathcal{Q} : ensemble fini d'états
- $\mathcal{I}(\subseteq \mathcal{Q})$: ensemble des états initiaux
- $\mathcal{F}(\subseteq \mathcal{Q})$: ensemble des états finaux
- Σ : alphabet
- $\delta \in \mathcal{P}(\mathcal{Q} \times \Sigma \times \mathcal{Q})$: relation de transition

Propriétés sympathiques (ou pas !)

⇒ Déterminisme

⇒ Minimalité

HLIN608 – Algorithmique du texte.

Alban MANCHERON

Alignement visuel de séquences

Recherche de motif

Extraction de motif

Contexte

Quelques programmes [presque] naïfs

Algorithme KMR

Structures d'index

Hachage

Arbres

Automates

Tables



Extraction de motif

Structures d'index [Automates]

Définition (Automate à états finis)

≡ quintuplet $(\mathcal{Q}, \mathcal{I}, \mathcal{F}, \Sigma, \delta)$

- \mathcal{Q} : ensemble fini d'états
- $\mathcal{I}(\subseteq \mathcal{Q})$: ensemble des états initiaux
- $\mathcal{F}(\subseteq \mathcal{Q})$: ensemble des états finaux
- Σ : alphabet
- $\delta \in \mathcal{P}(\mathcal{Q} \times \Sigma \times \mathcal{Q})$: relation de transition

Propriétés sympathiques (ou pas !)

- ⇒ Déterminisme
- ⇒ Cyclicité

⇒ Minimalité

HLIN608 – Algorithmique du texte.

Alban MANCHERON

Alignement visuel de séquences

Recherche de motif

Extraction de motif

Contexte

Quelques programmes [presque] naïfs

Algorithme KMR

Structures d'index

Hachage

Arbres

Automates

Tables

Extraction de motif

Structures d'index [Automates]

Définition (Automate à états finis)

≡ quintuplet $(\mathcal{Q}, \mathcal{I}, \mathcal{F}, \Sigma, \delta)$

- \mathcal{Q} : ensemble fini d'états
- $\mathcal{I}(\subseteq \mathcal{Q})$: ensemble des états initiaux
- $\mathcal{F}(\subseteq \mathcal{Q})$: ensemble des états finaux
- Σ : alphabet
- $\delta \in \mathcal{P}(\mathcal{Q} \times \Sigma \times \mathcal{Q})$: relation de transition

Propriétés sympathiques (ou pas !)

- ⇒ Déterminisme
- ⇒ Cyclicité

- ⇒ Minimalité
- ⇒ Homogénéité

HLIN608 – Algorithmique du texte.

Alban MANCHERON

Alignement visuel de séquences

Recherche de motif

Extraction de motif

Contexte

Quelques programmes [presque] naïfs

Algorithme KMR

Structures d'index

Hachage

Arbres

Automates

Tables



Extraction de motif

Structures d'index [Automates]

Définition (Automate à états finis)

≡ quintuplet $(\mathcal{Q}, \mathcal{I}, \mathcal{F}, \Sigma, \delta)$

- \mathcal{Q} : ensemble fini d'états
- $\mathcal{I}(\subseteq \mathcal{Q})$: ensemble des états initiaux
- $\mathcal{F}(\subseteq \mathcal{Q})$: ensemble des états finaux
- Σ : alphabet
- $\delta \in \mathcal{P}(\mathcal{Q} \times \Sigma \times \mathcal{Q})$: relation de transition

Propriétés sympathiques (ou pas !)

- | | |
|----------------|---------------|
| ⇒ Déterminisme | ⇒ Minimalité |
| ⇒ Cyclicité | ⇒ Homogénéité |

Automate \Leftrightarrow Langage

HLIN608 – Algorithmique du texte.

Alban MANCHERON

Alignement visuel de séquences

Recherche de motif

Extraction de motif

Contexte

Quelques programmes [presque] naïfs

Algorithme KMR

Structures d'index

Hachage

Arbres

Automates

Tables



Extraction de motif

Structures d'index [Automates]

Définition (Automate à états finis)

≡ quintuplet $(\mathcal{Q}, \mathcal{I}, \mathcal{F}, \Sigma, \delta)$

- \mathcal{Q} : ensemble fini d'états
- $\mathcal{I}(\subseteq \mathcal{Q})$: ensemble des états initiaux
- $\mathcal{F}(\subseteq \mathcal{Q})$: ensemble des états finaux
- Σ : alphabet
- $\delta \in \mathcal{P}(\mathcal{Q} \times \Sigma \times \mathcal{Q})$: relation de transition

Propriétés sympathiques (ou pas !)

- | | |
|----------------------------|---------------------------|
| \Rightarrow Déterminisme | \Rightarrow Minimalité |
| \Rightarrow Cyclicité | \Rightarrow Homogénéité |

Automate \Leftrightarrow Langage

- Automates des facteurs/suffixes

HLIN608 – Algorithmique du texte.

Alban MANCHERON

Alignement visuel de séquences

Recherche de motif

Extraction de motif

Contexte

Quelques programmes [presque] naïfs

Algorithme KMR

Structures d'index

Hachage

Arbres

Automates

Tables

Extraction de motif

Structures d'index [Automates]

Définition (Automate à états finis)

≡ quintuplet $(\mathcal{Q}, \mathcal{I}, \mathcal{F}, \Sigma, \delta)$

- \mathcal{Q} : ensemble fini d'états
- $\mathcal{I}(\subseteq \mathcal{Q})$: ensemble des états initiaux
- $\mathcal{F}(\subseteq \mathcal{Q})$: ensemble des états finaux
- Σ : alphabet
- $\delta \in \mathcal{P}(\mathcal{Q} \times \Sigma \times \mathcal{Q})$: relation de transition

Propriétés sympathiques (ou pas !)

- | | |
|----------------|---------------|
| ⇒ Déterminisme | ⇒ Minimalité |
| ⇒ Cyclicité | ⇒ Homogénéité |

Automate \Leftrightarrow Langage

- Automates des facteurs/suffixes
- Automates des sous-séquences

HLIN608 – Algorithmique du texte.

Alban MANCHERON

Alignement visuel de séquences

Recherche de motif

Extraction de motif

Contexte

Quelques programmes [presque] naïfs

Algorithme KMR

Structures d'index

Hachage

Arbres

Automates

Tables



Extraction de motif

Structures d'index [Automates]

Définition (Automate à états finis)

≡ quintuplet $(\mathcal{Q}, \mathcal{I}, \mathcal{F}, \Sigma, \delta)$

- \mathcal{Q} : ensemble fini d'états
- $\mathcal{I}(\subseteq \mathcal{Q})$: ensemble des états initiaux
- $\mathcal{F}(\subseteq \mathcal{Q})$: ensemble des états finaux
- Σ : alphabet
- $\delta \in \mathcal{P}(\mathcal{Q} \times \Sigma \times \mathcal{Q})$: relation de transition

Propriétés sympathiques (ou pas !)

- | | |
|----------------------------|---------------------------|
| \Rightarrow Déterminisme | \Rightarrow Minimalité |
| \Rightarrow Cyclicité | \Rightarrow Homogénéité |

Automate \Leftrightarrow Langage

- Automates des facteurs/suffixes
- Automates des sous-séquences
- Algorithmes (AHO-CORASIK, ...)

HLIN608 – Algorithmique du texte.

Alban MANCHERON

Alignement visuel de séquences

Recherche de motif

Extraction de motif

Contexte

Quelques programmes [presque] naïfs

Algorithmme KMR

Structures d'index

Hachage

Arbres

Automates

Tables



Extraction de motif

Structures d'index [Automates]

Définition (Automate à états finis)

= quintuplet $(Q, \Sigma, F, \delta, \sigma)$

- Q : ensemble fini d'états
- $\Sigma \subseteq Q$: ensemble des états initiaux
- $F \subseteq Q$: ensemble des états finaux
- Σ : alphabet
- $\delta \in \mathcal{P}(Q \times \Sigma \times Q)$: relation de transition

Automate des suffixes du mot
ROUDOUDOU

Propriétés sympathiques (ou pas !)

- ⇒ Déterminisme ⇒ Minimalité
⇒ Cyclicité ⇒ Homogénéité

Automate \Leftrightarrow Langage

- Automates des facteurs/suffixes
- Automates des sous-séquences
- Algorithmes (AHO-CORASIK, ...)

HLIN608 – Algorithmique du texte.

Alban MANCHERON

Alignement visuel de séquences

Recherche de motif

Extraction de motif

Contexte

Quelques programmes [presque] naïfs

Algorithmme KMR

Structures d'index

Hachage

Arbres

Automates

Tables

Extraction de motif

Structures d'index [Automates]

Définition (Automate à états finis)

quintuplet $(Q, \Sigma, F, \delta, q_0)$

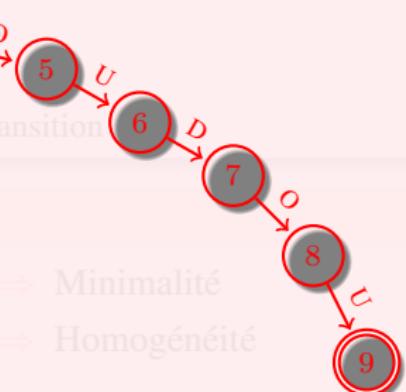
Automate des suffixes du mot ROUDOUDOU

- Q : ensemble fini d'états
- $I(\subseteq Q)$: ensemble des états initiaux
- $F(\subseteq Q)$: ensemble des états finaux
- Σ : alphabet
- $\delta \in \lambda^* \times \Sigma \times Q$: relation de transition

R → q_0 sympathiques (ou pas !)

⇒ Determinisme

⇒ Cyclicité



⇒ Minimalité

⇒ Homogénéité

Automate \Leftrightarrow Langage

- Automates des facteurs/suffixes
- Automates des sous-séquences
- Algorithmes (AHO-CORASIK, ...)

Extraction de motif

Structures d'index [Automates]



Définition (Automate à états finis)

quintuplet $(Q, \Sigma, F, \delta, q_0)$

Automate des suffixes du mot ROUDOUDOU

- Q : ensemble fini d'états
- $I(\subseteq Q)$: ensemble des états initiaux
- $F(\subseteq Q)$: ensemble des états finaux
- Σ : alphabet
- $\delta \in \Delta = \Sigma \times Q$: relation de transition



R → sympathiques (ou pas !)

⇒ Déterminisme

⇒ Cyclicité

⇒ Minimalité

⇒ Homogénéité

Automate ⇔ Langage

- Automates des facteurs/suffixes
- Automates des sous-séquences
- Algorithmes (AHO-CORASIK, ...)

HLIN608 – Algorithmique du texte.

Alban MANCHERON

Alignement visuel de séquences

Recherche de motif

Extraction de motif

Contexte

Quelques programmes [presque] naïfs

Algorithmme KMR

Structures d'index

Hachage

Arbres

Automates

Tables

Extraction de motif

Structures d'index [Automates]

Définition (Automate à états finis)

quintuplet $(Q, \Sigma, F, \delta, q_0)$

Automate des suffixes du mot
ROUDOUDOU

- Q : ensemble fini d'états
- $I(\subseteq Q)$: ensemble des états initiaux
- $F(\subseteq Q)$: ensemble des états finaux
- Σ : alphabet
- $\delta \in (\Sigma \times Q) \rightarrow Q$: relation de transition



Réflexion sympathiques (ou pas !)

⇒ Déterminisme

⇒ Cyclicité

⇒ Minimalité

⇒ Homogénéité

Automate \Leftrightarrow Langage

- Automates des facteurs/suffixes
- Automates des sous-séquences
- Algorithmes (AHO-CORASIK, ...)

Extraction de motif

Structures d'index [Automates]



HLIN608 – Algorithmique du texte.

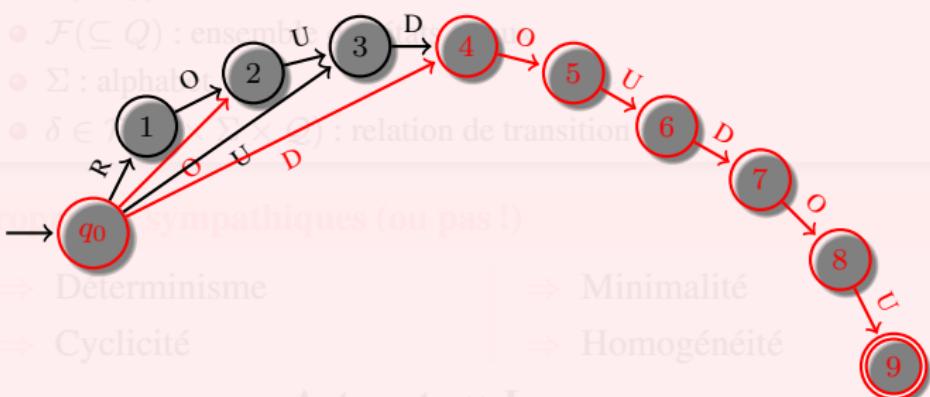
Alban MANCHERON

Définition (Automate à états finis)

quintuplet $(Q, \Sigma, F, \delta, q_0)$

Automate des suffixes du mot ROU**D**OU**D**OU**D**OU

- Q : ensemble fini d'états
- $I(\subseteq Q)$: ensemble des états initiaux
- $F(\subseteq Q)$: ensemble des états finaux
- Σ : alphabet
- $\delta \in \Delta = Q \times \Sigma$: relation de transition



- Automates des facteurs/suffixes
- Automates des sous-séquences
- Algorithmes (AHO-CORASIK, ...)

Alignement visuel de séquences

Recherche de motif

Extraction de motif

Contexte

Quelques programmes [presque] naïfs

Algorithmme KMR

Structures d'index

Hachage

Arbres

Automates

Tables

Extraction de motif

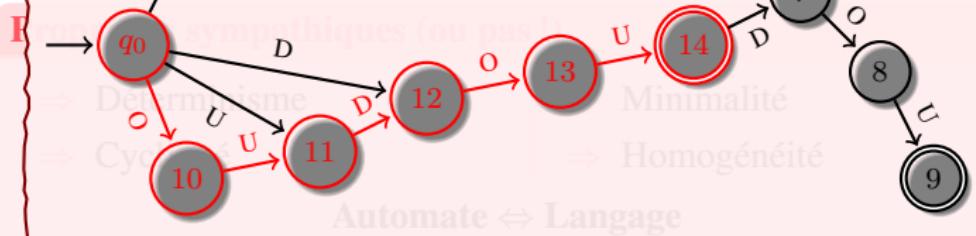
Structures d'index [Automates]

Définition (Automate à états finis)

quintuplet $(Q, \Sigma, F, \delta, q_0)$

Automate des suffixes du mot ROUDOUUDOU

- Q : ensemble fini d'états
- $I(\subseteq Q)$: ensemble des états initiaux
- $F(\subseteq Q)$: ensemble des états finaux
- Σ : alphabet
- $\delta \in \lambda^* \times \Sigma \times Q$: relation de transition



- Automates des facteurs/suffixes
- Automates des sous-séquences
- Algorithmes (AHO-CORASIK, ...)

Extraction de motif

Structures d'index [Automates]

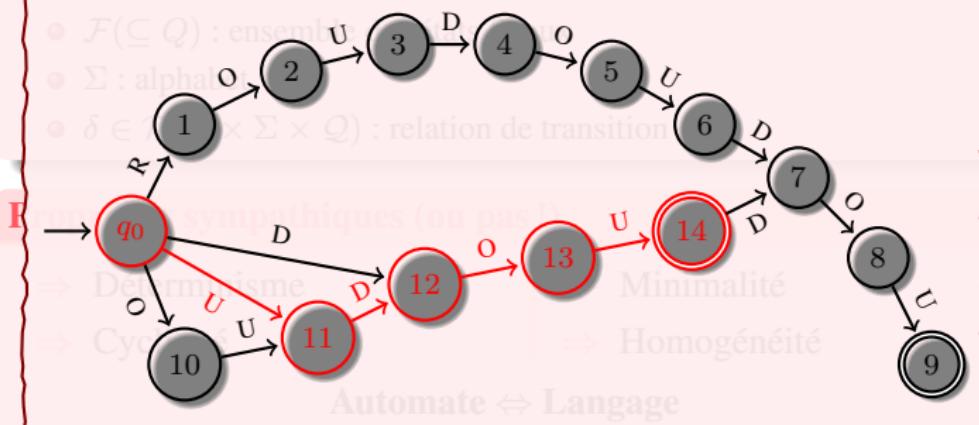


Définition (Automate à états finis)

quintuplet $(Q, \Sigma, F, \delta, q_0)$

Automate des suffixes du mot ROUDOUDOU

- Q : ensemble fini d'états
- $I(\subseteq Q)$: ensemble des états initiaux
- $F(\subseteq Q)$: ensemble des états finaux
- Σ : alphabet
- $\delta \in \lambda^* \times \Sigma \times Q$: relation de transition



- Automates des facteurs/suffixes
- Automates des sous-séquences
- Algorithmes (AHO-CORASIK, ...)

Alignement visuel de séquences

Recherche de motif

Extraction de motif

Contexte

Quelques programmes [presque] naïfs

Algorithmme KMR

Structures d'index

Hachage

Arbres

Automates

Tables

Extraction de motif

Structures d'index [Automates]

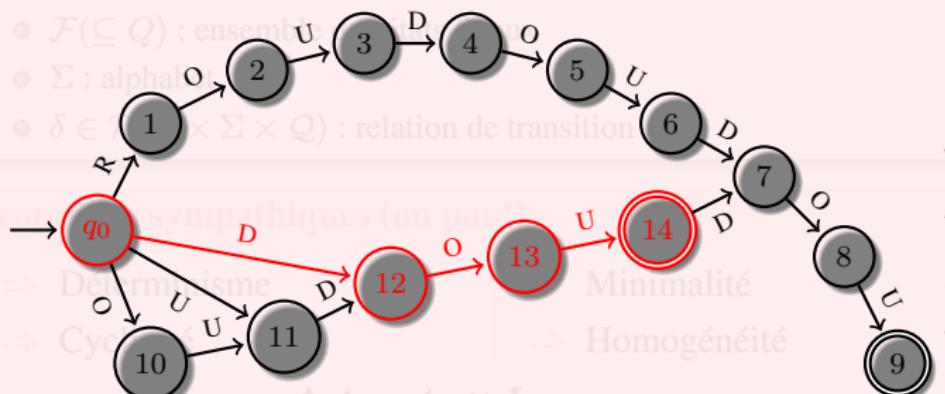
Définition (Automate à états finis)

quintuplet $(Q, \Sigma, F, \delta, q_0)$

Automate des suffixes du mot ROUDOUDOU

- Q : ensemble fini d'états
- $I(\subseteq Q)$: ensemble des états initiaux
- $F(\subseteq Q)$: ensemble des états finaux
- Σ : alphabet
- $\delta \in \lambda^* \times \Sigma \times Q$: relation de transition

R



Automate \Leftrightarrow Langage

- Automates des facteurs/suffixes
- Automates des sous-séquences
- Algorithmes (AHO-CORASIK, ...)

Extraction de motif

Structures d'index [Automates]



HLIN608 – Algorithmique du texte.

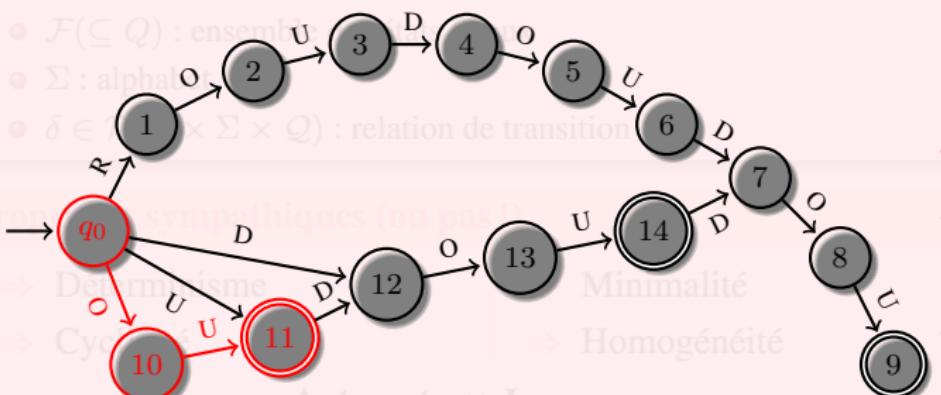
Alban MANCHERON

Définition (Automate à états finis)

quintuplet $(Q, \Sigma, F, \delta, q_0)$

Automate des suffixes du mot ROUDOUDOU

- Q : ensemble fini d'états
- $I(\subseteq Q)$: ensemble des états initiaux
- $F(\subseteq Q)$: ensemble des états finaux
- Σ : alphabet
- $\delta \in \lambda^* \times \Sigma \times Q$: relation de transition



- Automates des facteurs/suffixes
- Automates des sous-séquences
- Algorithmes (AHO-CORASIK, ...)

Alignement visuel de séquences

Recherche de motif

Extraction de motif

Contexte

Quelques programmes [presque] naïfs

Algorithmme KMR

Structures d'index

Hachage

Arbres

Automates

Tables

Extraction de motif

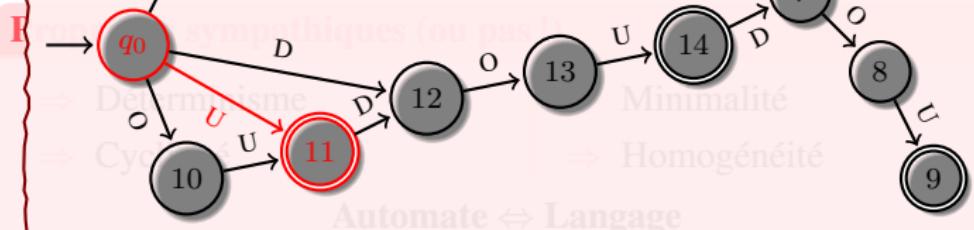
Structures d'index [Automates]

Définition (Automate à états finis)

quintuplet $(Q, \Sigma, F, \delta, q_0)$

Automate des suffixes du mot ROUDOUDOU

- Q : ensemble fini d'états
- $I(\subseteq Q)$: ensemble des états initiaux
- $F(\subseteq Q)$: ensemble des états finaux
- Σ : alphabet
- $\delta \in \lambda^* \times \Sigma \times Q$: relation de transition



- Automates des facteurs/suffixes
- Automates des sous-séquences
- Algorithmes (AHO-CORASIK, ...)

Extraction de motif

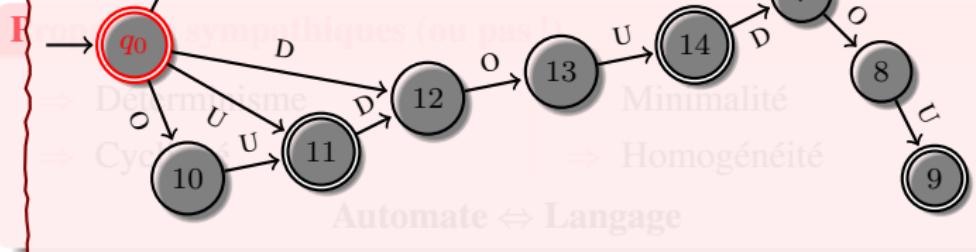
Structures d'index [Automates]

Définition (Automate à états finis)

quintuplet $(Q, \Sigma, F, \delta, q_0)$

Automate des suffixes du mot
ROUDOUDOU

- Q : ensemble fini d'états
- $I(\subseteq Q)$: ensemble des états initiaux
- $F(\subseteq Q)$: ensemble des états finaux
- Σ : alphabet
- $\delta \in \lambda^* \times \Sigma \times Q$: relation de transition



- Automates des facteurs/suffixes
- Automates des sous-séquences
- Algorithmes (AHO-CORASIK, ...)



Extraction de motif

Structures d'index [Tables]

Définition (Table des suffixes)

≡ tableau des positions des suffixes d'un mot dans l'ordre lexicographique.

HLIN608 – Algorithmique du texte.

Alban MANCHERON

Alignement visuel de séquences

Recherche de motif

Extraction de motif

Contexte

Quelques programmes [presque] naïfs

Algorithmme KMR

Structures d'index

Hachage

Arbres

Automates

Tables



Extraction de motif

Structures d'index [Tables]

Définition (Table des suffixes)

≡ tableau des positions des suffixes d'un mot dans l'ordre lexicographique.



HLIN608 – Algorithmique du texte.

Alban MANCHERON

Alignement visuel de séquences

Recherche de motif

Extraction de motif

Contexte

Quelques programmes [presque] naïfs

Algorithmme KMR

Structures d'index

Hachage

Arbres

Automates

Tables



Extraction de motif

Structures d'index [Tables]

Définition (Table des suffixes)

≡ tableau des positions des suffixes d'un mot dans l'ordre lexicographique.

+

Définition (Table des plus long préfixes communs)

≡ longueur du plus long préfixe commun entre deux suffixes consécutifs de la table des suffixes.

⇒ Amélioration des performances de recherche

HLIN608 – Algorithmique du texte.

Alban MANCHERON

Alignement visuel de séquences

Recherche de motif

Extraction de motif

Contexte

Quelques programmes [presque] naïfs

Algorithmme KMR

Structures d'index

Hachage

Arbres

Automates

Tables



Extraction de motif

Structures d'index [Tables]

Définition (Table des suffixes)

tableau des positions des suffixes d'un mot dans l'ordre lexicographique.

Table des suffixes du mot
ROUDOUDOU\$



Définition (Table des plus long préfixes communs)

longueur du plus long préfixe commun entre deux suffixes consécutifs de la table des suffixes.

Amélioration des performances de recherche

HLIN608 – Algorithmique du texte.

Alban MANCHERON

Alignement visuel de séquences

Recherche de motif

Extraction de motif

Contexte

Quelques programmes [presque] naïfs

Algorithmme KMR

Structures d'index

Hachage

Arbres

Automates

Tables



Extraction de motif

Structures d'index [Tables]

Définition (Table des suffixes)

tableau des positions des suffixes dans l'ordre lexicographique.

Table des suffixes du mot
ROUDOUDOU\$

1 ROUDOUDOU\$

+

Définition (Table des plus long préfixes communs)

longueur du plus long préfixe commun entre deux suffixes consécutifs de la table des suffixes.

Amélioration des performances de recherche

HLIN608 – Algorithmique du texte.

Alban MANCHERON

Alignement visuel de séquences

Recherche de motif

Extraction de motif

Contexte

Quelques programmes [presque] naïfs

Algorithmme KMR

Structures d'index

Hachage

Arbres

Automates

Tables



Extraction de motif

Structures d'index [Tables]

Définition (Table des suffixes)

tableau des positions des suffixes dans l'ordre lexicographique.

Table des suffixes du mot
ROUDOUDOU\$

- | | |
|---|-------------|
| 1 | ROUDOUDOU\$ |
| 2 | OUDOUDOU\$ |

+

Illustration (Table des plus long préfixes communs)

longueur du plus long préfixe commun entre deux suffixes consécutifs de la table des suffixes.

Amélioration des performances de recherche

HLIN608 – Algorithmique du texte.

Alban MANCHERON

Alignement visuel de séquences

Recherche de motif

Extraction de motif

Contexte

Quelques programmes [presque] naïfs

Algorithmme KMR

Structures d'index

Hachage

Arbres

Automates

Tables

Extraction de motif

Structures d'index [Tables]

Définition (Table des suffixes)

tableau des positions des suffixes dans l'ordre lexicographique.

Table des suffixes du mot
ROUDOUUDOU\$

1	ROUDOUUDOU\$
2	OUDOUUDOU\$
3	UDOUDOU\$

+

longueur du plus long préfixe commun entre deux suffixes consécutifs de la table des suffixes.

Amélioration des performances de recherche

HLIN608 – Algorithmique du texte.

Alban MANCHERON

Alignement visuel de séquences

Recherche de motif

Extraction de motif

Contexte

Quelques programmes [presque] naïfs

Algorithmme KMR

Structures d'index

Hachage

Arbres

Automates

Tables



Extraction de motif

Structures d'index [Tables]

Définition (Table des suffixes)

tableau des positions des suffixes et dans l'ordre lexicographique.

Table des suffixes du mot
ROUDOUDOU\$

1	ROUDOUDOU\$
2	OUDOUDOU\$
3	UDOUDOU\$
4	DOUDOU\$

+

Identifie les plus long préfixes communs)

Identifie le plus long préfixe commun entre deux suffixes consécutifs de la table des suffixes.

Amélioration des performances de recherche

HLIN608 – Algorithmique du texte.

Alban MANCHERON

Alignement visuel de séquences

Recherche de motif

Extraction de motif

Contexte

Quelques programmes [presque] naïfs

Algorithmme KMR

Structures d'index

Hachage

Arbres

Automates

Tables



Extraction de motif

Structures d'index [Tables]

Définition (Table des suffixes)

Table des suffixes du mot
ROUDOU\$

1	ROUDO\$
2	OUDO\$
3	UDO\$
4	DO\$
5	O\$

+

Identification des plus long préfixes communs)

Identification des plus long préfixe commun entre deux suffixes

Conseil : Utiliser la table des suffixes.

Amélioration des performances de recherche

HLIN608 – Algorithmique du texte.

Alban MANCHERON

Alignement visuel de séquences

Recherche de motif

Extraction de motif

Contexte

Quelques programmes [presque] naïfs

Algorithmme KMR

Structures d'index

Hachage

Arbres

Automates

Tables



Extraction de motif

Structures d'index [Tables]

Définition (Table des suffixes)

Table des suffixes du mot
ROUDOUDOU\$

1	ROUDOUDOU\$
2	OUDOUDOU\$
3	UDOUDOU\$
4	DOUDOU\$
5	OUDOU\$
6	UDOU\$

+

Identifie les plus long préfixes communs)

Identifie le plus long préfixe commun entre deux suffixes

Construit à l'aide de la table des suffixes.

Amélioration des performances de recherche

HLIN608 – Algorithmique du texte.

Alban MANCHERON

Alignement visuel de séquences

Recherche de motif

Extraction de motif

Contexte

Quelques programmes [presque] naïfs

Algorithmme KMR

Structures d'index

Hachage

Arbres

Automates

Tables



Extraction de motif

Structures d'index [Tables]

Définition (Table des suffixes)

Table des suffixes du mot
ROUDOUDOU\$

1	ROUDOUDOU\$
2	OUDOUDOU\$
3	UDOUDOU\$
4	DOUDOU\$
5	OUDOU\$
6	UDOU\$
7	DOU\$

+

Identifie les plus long préfixes communs)

Identifie les plus long préfixe commun entre deux suffixes

Construit la table des suffixes.

Améliore les performances de recherche

HLIN608 – Algorithmique du texte.

Alban MANCHERON

Alignement visuel de séquences

Recherche de motif

Extraction de motif

Contexte

Quelques programmes [presque] naïfs

Algorithmme KMR

Structures d'index

Hachage

Arbres

Automates

Tables



Extraction de motif

Structures d'index [Tables]

Définition (Table des suffixes)

Table des suffixes du mot
ROUDOUDOU\$

1	ROUDOUDOU\$
2	OUDOUDOU\$
3	UDOUDOU\$
4	DOUDOU\$
5	OUDOU\$
6	UDOU\$
7	DOU\$
8	OU\$

+

Identifie les plus long préfixes communs)

Identifie le plus long préfixe commun entre deux suffixes

Construit la table des suffixes.

Améliore les performances de recherche

HLIN608 – Algorithmique du texte.

Alban MANCHERON

Alignement visuel de séquences

Recherche de motif

Extraction de motif

Contexte

Quelques programmes [presque] naïfs

Algorithmme KMR

Structures d'index

Hachage

Arbres

Automates

Tables

Extraction de motif

Structures d'index [Tables]

Définition (Table des suffixes)

Table des suffixes du mot
ROUDOUDOOU\$

1	ROUDOUDOOU\$
2	OUDOUDOOU\$
3	UDOUDOOU\$
4	DOUDOOU\$
5	OUDOU\$
6	UDOU\$
7	DOU\$
8	OU\$
9	U\$

+

Identifie les plus long préfixes communs)

Identifie le plus long préfixe commun entre deux suffixes

Construit la table des suffixes.

Améliore les performances de recherche

HLIN608 – Algorithmique du texte.

Alban MANCHERON

Alignement visuel de séquences

Recherche de motif

Extraction de motif

Contexte

Quelques programmes [presque] naïfs

Algorithmme KMR

Structures d'index

Hachage

Arbres

Automates

Tables

Extraction de motif

Structures d'index [Tables]

Définition (Table des suffixes)

Table des suffixes du mot
ROUDOUDOU\$

1	ROUDOUDOU\$
2	OUDOUDOU\$
3	UDOUDOU\$
4	DOUDOU\$
5	OUDOU\$
6	UDOU\$
7	DOU\$
8	OU\$
9	U\$
10	\$

+

Identifie les plus long préfixes communs)

Identifie les plus long préfixe commun entre deux suffixes

Construit la table des suffixes.

Améliore les performances de recherche

HLIN608 – Algorithmique du texte.

Alban MANCHERON

Alignement visuel de séquences

Recherche de motif

Extraction de motif

Contexte

Quelques programmes [presque] naïfs

Algorithmme KMR

Structures d'index

Hachage

Arbres

Automates

Tables



Extraction de motif

Structures d'index [Tables]

Définition (Table des suffixes)

Table des suffixes du mot
ROUDOUDOU\$

1	ROUDOUDOU\$
2	OUDOUDOU\$
3	UDOUDOU\$
4	DOUDOU\$
5	OUDOU\$
6	UDOU\$
7	DOU\$
8	OU\$
9	U\$
10	\$

+

Identifie les plus long préfixes communs)

Identifie les plus long préfixe commun entre deux suffixes

Construit la table des suffixes.

Améliore les performances de recherche

HLIN608 – Algorithmique du texte.

Alban MANCHERON

Alignement visuel de séquences

Recherche de motif

Extraction de motif

Contexte

Quelques programmes [presque] naïfs

Algorithmme KMR

Structures d'index

Hachage

Arbres

Automates

Tables

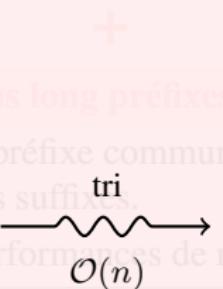
Extraction de motif

Structures d'index [Tables]

Définition (Table des suffixes)

Table des suffixes du mot
ROUDOUDOU\$

1	ROUDOUDOU\$
2	OUDOUDOU\$
3	UDOUDOU\$
4	DOUDOU\$
5	OUDOU\$
6	UDOU\$
7	DOU\$
8	OU\$
9	U\$
10	\$



HLIN608 – Algorithmique du texte.

Alban MANCHERON

Alignement visuel de séquences

Recherche de motif

Extraction de motif

Contexte

Quelques programmes [presque] naïfs

Algorithmme KMR

Structures d'index

Hachage

Arbres

Automates

Tables

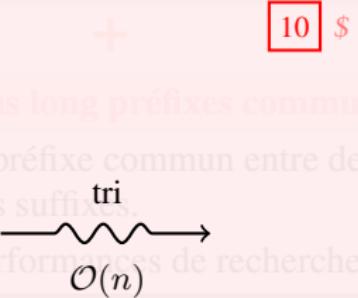
Extraction de motif

Structures d'index [Tables]

Définition (Table des suffixes)

Table des suffixes du mot
ROUDOUDOU\$

1	ROUDOUDOU\$
2	OUDOUDOU\$
3	UDOUDOU\$
4	DOUDOU\$
5	OUDOU\$
6	UDOU\$
7	DOU\$
8	OU\$
9	U\$
10	\$



HLIN608 – Algorithmique du texte.

Alban MANCHERON

Alignement visuel de séquences

Recherche de motif

Extraction de motif

Contexte

Quelques programmes [presque] naïfs

Algorithmme KMR

Structures d'index

Hachage

Arbres

Automates

Tables

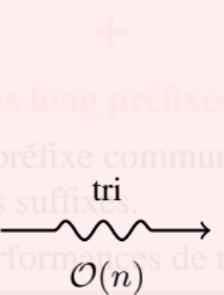
Extraction de motif

Structures d'index [Tables]

Définition (Table des suffixes)

Table des suffixes du mot
ROUDOUDOU\$

1	ROUDOUDOU\$
2	OUDOUDOU\$
3	UDOUDOU\$
4	DOUDOU\$
5	OUDOU\$
6	UDOU\$
7	DOU\$
8	OU\$
9	U\$
10	\$



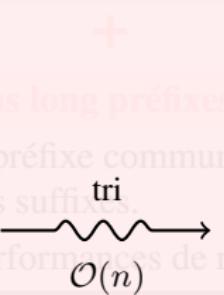
Extraction de motif

Structures d'index [Tables]

Définition (Table des suffixes)

Table des suffixes du mot
ROUDOUDOU\$

1	ROUDOUDOU\$
2	OUDOUDOU\$
3	UDOUDOU\$
4	DOUDOU\$
5	OUDOU\$
6	UDOU\$
7	DOU\$
8	OU\$
9	U\$
10	\$



10	\$
7	DOU\$
4	DOUDOU\$

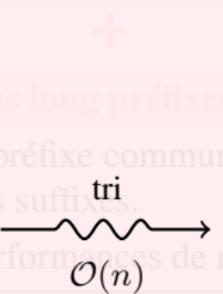
Extraction de motif

Structures d'index [Tables]

Définition (Table des suffixes)

Table des suffixes du mot
ROUDOUDOU\$

1	ROUDOUDOU\$
2	OUDOUDOU\$
3	UDOUDOU\$
4	DOUDOU\$
5	OUDOU\$
6	UDOU\$
7	DOU\$
8	OU\$
9	U\$
10	\$



10	\$
7	DOU\$
4	DOUDOU\$
8	OU\$

Extraction de motif

Structures d'index [Tables]

Définition (Table des suffixes)

Table des suffixes du mot
ROUDOUDOU\$

1	ROUDOUDOU\$	+	10	\$
2	OUDOUDOU\$		7	DOU\$
3	UDOUDOU\$		4	DOUDOU\$
4	DOUDOU\$		8	OU\$
5	OUDOU\$	tri	5	OUDOU\$
6	UDOU\$			
7	DOU\$			
8	OU\$			
9	U\$			
10	\$			

$\mathcal{O}(n)$

Extraction de motif

Structures d'index [Tables]

Définition (Table des suffixes)

Table des suffixes du mot
ROUDOUDOU\$

1	ROUDOUDOU\$	+	10	\$
2	OUDOUDOU\$		7	DOU\$
3	UDOUDOU\$		4	DOUDOU\$
4	DOUDOU\$		8	OU\$
5	OUDOU\$	tri	5	OUDOU\$
6	UDOU\$		2	OUDOUDOU\$
7	DOU\$			
8	OU\$			
9	U\$			
10	\$			

HLIN608 – Algorithmique du texte.

Alban MANCHERON

Alignement visuel de séquences

Recherche de motif

Extraction de motif

Contexte

Quelques programmes [presque] naïfs

Algorithme KMR

Structures d'index

Hachage

Arbres

Automates

Tables

Extraction de motif

Structures d'index [Tables]

Définition (Table des suffixes)

Table des suffixes du mot
ROUDOUDOU\$

1	ROUDOUDOU\$	+	10	\$
2	OUDOUDOU\$		7	DOU\$
3	UDOUDOU\$		4	DOUDOU\$
4	DOUDOU\$		8	OU\$
5	OUDOU\$	tri	5	OUDOU\$
6	UDOU\$		2	OUDOUDOU\$
7	DOU\$		1	ROUDOUDOU\$
8	OU\$			
9	U\$			
10	\$			

HLIN608 – Algorithmique du texte.

Alban MANCHERON

Alignement visuel de séquences

Recherche de motif

Extraction de motif

Contexte

Quelques programmes [presque] naïfs

Algorithmme KMR

Structures d'index

Hachage

Arbres

Automates

Tables

Extraction de motif

Structures d'index [Tables]

Définition (Table des suffixes)

Table des suffixes du mot
ROUDOUDOU\$

1	ROUDOUDOU\$	+	10	\$
2	OUDOUDOU\$		7	DOU\$
3	UDOUDOU\$		4	DOUDOU\$
4	DOUDOU\$		8	OU\$
5	OUDOU\$	tri	5	OUDOU\$
6	UDOU\$		2	OUDOUDOU\$
7	DOU\$		1	ROUDOUDOU\$
8	OU\$		9	U\$
9	U\$			
10	\$			

HLIN608 – Algorithmique du texte.

Alban MANCHERON

Alignement visuel de séquences

Recherche de motif

Extraction de motif

Contexte

Quelques programmes [presque] naïfs

Algorithmme KMR

Structures d'index

Hachage

Arbres

Automates

Tables

Extraction de motif

Structures d'index [Tables]

Définition (Table des suffixes)

Table des suffixes du mot
ROUDOUDOU\$

1	ROUDOUDOU\$	+	10	\$
2	OUDOUDOU\$		7	DOU\$
3	UDOUDOU\$		4	DOUDOU\$
4	DOUDOU\$		8	OU\$
5	OUDOU\$	tri	5	OUDOU\$
6	UDOOU\$	\rightarrow	2	OUDOUDOU\$
7	DOU\$		1	ROUDOUDOU\$
8	OU\$		9	U\$
9	U\$		6	UDOOU\$
10	\$			

HLIN608 – Algorithmique du texte.

Alban MANCHERON

Alignement visuel de séquences

Recherche de motif

Extraction de motif

Contexte

Quelques programmes [presque] naïfs

Algorithmme KMR

Structures d'index

Hachage

Arbres

Automates

Tables

Extraction de motif

Structures d'index [Tables]

Définition (Table des suffixes)

Table des suffixes du mot
ROUDOUDOU\$

1	ROUDOUDOU\$	+	10	\$
2	OUDOUDOU\$		7	DOU\$
3	UDOUDOU\$		4	DOUDOU\$
4	DOUDOU\$		8	OU\$
5	OUDOU\$	tri	5	OUDOU\$
6	UDOOU\$		2	OUDOUDOU\$
7	DOU\$		1	ROUDOUDOU\$
8	OU\$		9	U\$
9	U\$		6	UDOOU\$
10	\$		3	UDOUDOU\$

Extraction de motif

Structures d'index [Tables]

Définition (Table des suffixes)

Table des suffixes du mot
ROUDOUDOU\$

1	ROUDOUDOU\$	+	10	\$
2	OUDOUDOU\$		7	DOU\$
3	UDOUDOU\$		4	DOUDOU\$
4	DOUDOU\$		8	OU\$
5	OUDOU\$	tri	5	OUDOU\$
6	UDOOU\$		2	OUDOUDOU\$
7	DOU\$		1	ROUDOUDOU\$
8	OU\$		9	U\$
9	U\$		6	UDOOU\$
10	\$		3	UDOUDOU\$

HLIN608 – Algorithmique du texte.

Alban MANCHERON

Alignement visuel de séquences

Recherche de motif

Extraction de motif

Contexte

Quelques programmes [presque] naïfs

Algorithmme KMR

Structures d'index

Hachage

Arbres

Automates

Tables

Extraction de motif

Structures d'index [Tables]

Définition (Table des suffixes)

tableau des positions des suffixes et dans l'ordre lexicographique.

Table des suffixes du mot
ROUDOUDOU\$



Définition (Table des plus long préfixes)

longueur du plus long préfixe commun entre deux suffixes consécutifs de la table des suffixes

Amélioration des performances de recherche

10	\$
7	DOU\$
4	DOUDOU\$
8	OU\$
5	OUDOU\$
2	OUDOUDOU\$
1	ROUDOUDOU\$
9	U\$
6	UDOUDOU\$
3	UDOUDOU\$

Extraction de motif

Structures d'index [Tables]

Définition (Table des suffixes)

tableau des positions des suffixes et dans l'ordre lexicographique.

Table des suffixes du mot
ROUDOUDOU\$

1 0

+

Définition (Table des plus long préfixes)

longueur du plus long préfixe commun entre deux suffixes consécutifs de la table des suffixes

LCP
← →

Amélioration des performances de recherche

10	\$
7	DOU\$
4	DOUDOU\$
8	OU\$
5	OUDOU\$
2	OUDOUDOU\$
1	ROUDOUDOU\$
9	U\$
6	UDOUDOU\$
3	UDOUDOU\$

Extraction de motif

Structures d'index [Tables]

Définition (Table des suffixes)

tableau des positions des suffixes et dans l'ordre lexicographique.

Table des suffixes du mot
ROUDOUDOU\$

1	0
2	0

+

10	\$
7	DOU\$
4	DOUDOU\$
8	OU\$
5	OUDOU\$
2	OUDOUDOU\$
1	ROUDOUDOU\$
9	U\$
6	UDOUDOU\$
3	UDOUDOU\$

Identification (Table des plus long préfixes)

longueur du plus long préfixe commun entre deux suffixes consécutifs de la table des suffixes

LCP

Amélioration des performances de recherche

Extraction de motif

Structures d'index [Tables]

Définition (Table des suffixes)

Table des suffixes du mot
ROUDOUDOU\$

1	0
2	0
3	3

Ideas (Table des plus long préfixes)



10	\$
7	DOU\$
4	DOUDOU\$
8	OU\$
5	OUDOU\$
2	OUDOUDOU\$
1	ROUDOUDOU\$
9	U\$
6	UDOUDOU\$
3	UDOUDOU\$

Extraction de motif

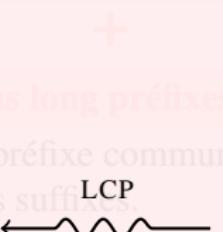
Structures d'index [Tables]

Définition (Table des suffixes)

Table des suffixes du mot
ROUDOUDOU\$

10	\$
7	DOU\$
4	DOUDOU\$
8	OU\$
5	OUDOU\$
2	OUDOUDOU\$
1	ROUDOUDOU\$
9	U\$
6	UDOUDOU\$
3	UDOUDOU\$

1	0
2	0
3	3
4	0



Ideas (Table des plus long préfixes)

longueur du plus long préfixe commun entre deux suffixes

consecutifs de la table des suffixes

Amélioration des performances de recherche

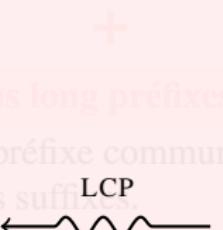
Extraction de motif

Structures d'index [Tables]

Définition (Table des suffixes)

Table des suffixes du mot
ROUDOUDOU\$

1	0
2	0
3	3
4	0
5	2



10	\$
7	DOU\$
4	DOUDOU\$
8	OU\$
5	OUDOU\$
2	OUDOUDOU\$
1	ROUDOUDOU\$
9	U\$
6	UDOUDOU\$
3	UDOUDOU\$

Extraction de motif

Structures d'index [Tables]

Définition (Table des suffixes)

Table des suffixes du mot
ROUDOUDOU\$

1	0
2	0
3	3
4	0
5	2
6	5



10	\$
7	DOU\$
4	DOUDOU\$
8	OU\$
5	OUDOU\$
2	OUDOUDOU\$
1	ROUDOUDOU\$
9	U\$
6	UDOUDOU\$
3	UDOUDOU\$

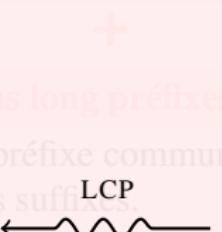
Extraction de motif

Structures d'index [Tables]

Définition (Table des suffixes)

Table des suffixes du mot
ROUDOUDOU\$

1	0
2	0
3	3
4	0
5	2
6	5
7	0



10	\$
7	DOU\$
4	DOUDOU\$
8	OU\$
5	OUDOU\$
2	OUDOUDOU\$
1	ROUDOUDOU\$
9	U\$
6	UDOUDOU\$
3	UDOUDOU\$

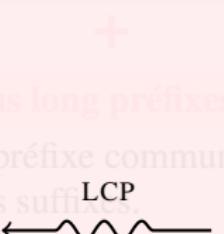
Extraction de motif

Structures d'index [Tables]

Définition (Table des suffixes)

Table des suffixes du mot
ROUDOUDOU\$

1	0
2	0
3	3
4	0
5	2
6	5
7	0
8	0



10	\$
7	DOU\$
4	DOUDOU\$
8	OU\$
5	OUDOU\$
2	OUDOUDOU\$
1	ROUDOUDOU\$
9	U\$
6	UDOUDOU\$
3	UDOUDOU\$

Extraction de motif

Structures d'index [Tables]

Définition (Table des suffixes)

Table des suffixes du mot
ROUDOUDOU\$

1	0
2	0
3	3
4	0
5	2
6	5
7	0
8	0
9	1



10	\$
7	DOU\$
4	DOUDOU\$
8	OU\$
5	OUDOU\$
2	OUDOUDOU\$
1	ROUDOUDOU\$
9	U\$
6	UDOUDOU\$
3	UDOUDOU\$

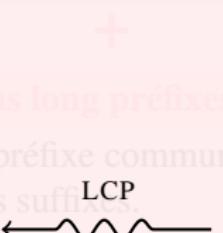
Extraction de motif

Structures d'index [Tables]

Définition (Table des suffixes)

Table des suffixes du mot
ROUDOUDOU\$

1	0
2	0
3	3
4	0
5	2
6	5
7	0
8	0
9	1
10	4



10	\$
7	DOU\$
4	DOUDOU\$
8	OU\$
5	OUDOU\$
2	OUDOUDOU\$
1	ROUDOUDOU\$
9	U\$
6	UDOU\$
3	UDOUDOU\$

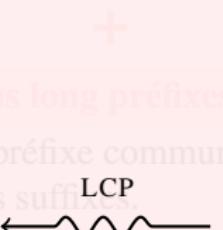
Extraction de motif

Structures d'index [Tables]

Définition (Table des suffixes)

Table des suffixes du mot
ROUDOUDOU\$

1	0
2	0
3	3
4	0
5	2
6	5
7	0
8	0
9	1
10	4



10	\$
7	DOU\$
4	DOUDOU\$
8	OU\$
5	OUDOU\$
2	OUDOUDOU\$
1	ROUDOUDOU\$
9	U\$
6	UDOUDOU\$
3	UDOUDOU\$

Extraction de motif

Structures d'index [Tables]

Définition (Table des suffixes)

≡ tableau des positions des suffixes d'un mot dans l'ordre lexicographique.



Définition (Table des plus long préfixes communs)

≡ longueur du plus long préfixe commun entre deux suffixes consécutifs de la table des suffixes.

⇒ Amélioration des performances de recherche

Définition (FM-index)

Le FM-index (Ferragina–Manzini index) est la structure résultant de la compression de la transformée de BURROWS-WHEELER. Sa construction est basée sur la table des suffixes pour le calcul de la transformée.



Extraction de motif

Structures d'index [Tables]

Définition (Table des suffixes)

Transformée de BURROWS-WHEELER du mot
ROUDOUDOU\$

R O U D O U D O U \$ +

Définition (Table des plus long préfixes communs)

longueur du plus long préfixe commun entre deux suffixes consécutifs de la table des suffixes.

Amélioration des performances de recherche

Définition (FM-index)

Le FM-index (Ferragina–Manzini index) est la structure résultante de la compression de la transformée de BURROWS-WHEELER. Sa construction est basée sur la table des suffixes pour le calcul de la transformée.

HLIN608 – Algorithmique du texte.

Alban MANCHERON

Alignement visuel de séquences

Recherche de motif

Extraction de motif

Contexte

Quelques programmes [presque] naïfs

Algorithmme KMR

Structures d'index

Hachage

Arbres

Automates

Tables

Extraction de motif

Structures d'index [Tables]

Définition (Table des suffixes)

Transformée de BURROWS-WHEELER du mot
ROUDOUDOU\$

R O U D O U D O U \$ +
\$ R O U D O U D O U

Définition (Table des plus long préfixes communs)

longueur du plus long préfixe commun entre deux suffixes consécutifs de la table des suffixes.

Amélioration des performances de recherche

Définition (FM-index)

Le FM-index (Ferragina–Manzini index) est la structure résultante de la compression de la transformée de BURROWS-WHEELER. Sa construction est basée sur la table des suffixes pour le calcul de la transformée.

HLIN608 – Algorithmique du texte.

Alban MANCHERON

Alignement visuel de séquences

Recherche de motif

Extraction de motif

Contexte

Quelques programmes [presque] naïfs

Algorithmme KMR

Structures d'index

Hachage

Arbres

Automates

Tables

Extraction de motif

Structures d'index [Tables]

Définition (Table des suffixes)

Transformée de BURROWS-WHEELER du mot
ROUDOUDOU\$

R O U D O U D O U \$
\$ R O U D O U D O U
U \$ R O U D O U D O

+

(longueur du plus long préfixe commun)

longueur du plus long préfixe commun entre deux suffixes consécutifs de la table des suffixes.

Amélioration des performances de recherche

Définition (FM-index)

Le FM-index (Ferragina–Manzini index) est la structure résultante de la compression de la transformée de BURROWS-WHEELER. Sa construction est basée sur la table des suffixes pour le calcul de la transformée.

HLIN608 – Algorithmique du texte.

Alban MANCHERON

Alignement visuel de séquences

Recherche de motif

Extraction de motif

Contexte

Quelques programmes [presque] naïfs

Algorithmme KMR

Structures d'index

Hachage

Arbres

Automates

Tables

Extraction de motif

Structures d'index [Tables]

Définition (Table des suffixes)

Transformée de BURROWS-WHEELER du mot
ROUDOUDOU\$

R O U D O U D O U \$
\$ R O U D O U D O U
U \$ R O U D O U D O
O U \$ R O U D O U D

+

(Ensuite, on recherche les longs préfixes communs)

longueur du plus long préfixe commun entre deux suffixes consécutifs de la table des suffixes.

Amélioration des performances de recherche

Définition (FM-index)

Le FM-index (Ferragina–Manzini index) est la structure résultante de la compression de la transformée de BURROWS-WHEELER.

Sa construction est basée sur la table des suffixes pour le calcul de la transformée.

HLIN608 – Algorithmique du texte.

Alban MANCHERON

Alignement visuel de séquences

Recherche de motif

Extraction de motif

Contexte

Quelques programmes [presque] naïfs

Algorithmme KMR

Structures d'index

Hachage

Arbres

Automates

Tables

Extraction de motif

Structures d'index [Tables]

Définition (Table des suffixes)

Transformée de BURROWS-WHEELER du mot
ROUDOUDOU\$

R O U D O U D O U \$ +
\$ R O U D O U D O U
U \$ R O U D O U D O
O U \$ R O U D O U D
D O U \$ R O U D O U

(Ensuite, on recherche les longs préfixes communs)

long suffixe commun entre deux suffixes
consecutifs de la table des suffixes.

Amélioration des performances de recherche

Définition (FM-index)

Le FM-index (Ferragina–Manzini index) est la structure résultante de la compression de la transformée de BURROWS-WHEELER.

Sa construction est basée sur la table des suffixes pour le calcul de la transformée.

HLIN608 – Algorithmique du texte.

Alban MANCHERON

Alignement visuel de séquences

Recherche de motif

Extraction de motif

Contexte

Quelques programmes [presque] naïfs

Algorithmes KMR

Structures d'index

Hachage

Arbres

Automates

Tables

Extraction de motif

Structures d'index [Tables]

Définition (Table des suffixes)

Transformée de BURROWS-WHEELER du mot
ROUDOUDOU\$

R O U D O U D O U \$
\$ R O U D O U D O U
U \$ R O U D O U D O
O U \$ R O U D O U D
D O U \$ R O U D O U
U D O U \$ R O U D O

+

(Ensuite on recherche les longs préfixes communs)

long suffixe commun entre deux suffixes
consécutifs dans la transformée.

Amélioration des performances de recherche

Définition (FM-index)

Le FM-index (Ferragina–Manzini index) est la structure résultante de la compression de la transformée de BURROWS-WHEELER.
Sa construction est basée sur la table des suffixes pour le calcul de la transformée.

HLIN608 – Algorithmique du texte.

Alban MANCHERON

Alignement visuel de séquences

Recherche de motif

Extraction de motif

Contexte

Quelques programmes [presque] naïfs

Algorithmie KMR

Structures d'index

Hachage

Arbres

Automates

Tables

Extraction de motif

Structures d'index [Tables]

Définition (Table des suffixes)

Transformée de BURROWS-WHEELER du mot
ROUDOUDOU\$

R O U D O U D O U \$ +
\$ R O U D O U D O U
U \$ R O U D O U D O (long préfixes communs)
O U \$ R O U D O U D
D O U \$ R O U D O U
U D O U \$ R O U D O
O U D O U \$ R O U D

Définition (FM-index)

Le FM-index (Ferragina–Manzini index) est la structure résultante de la compression de la transformée de BURROWS-WHEELER. Sa construction est basée sur la table des suffixes pour le calcul de la transformée.

Extraction de motif

Structures d'index [Tables]

Définition (Table des suffixes)

Transformée de BURROWS-WHEELER du mot
ROUDOUDOU\$

R O U D O U D O U \$ +
\$ R O U D O U D O U
U \$ R O U D O U D O
O U \$ R O U D O U D
D O U \$ R O U D O U
U D O U \$ R O U D O
O U D O U \$ R O U D
D O U D O U \$ R O U

Définition (FM-index)

Le FM-index (Ferragina–Manzini index) est la structure résultante de la compression de la transformée de BURROWS-WHEELER. Sa construction est basée sur la table des suffixes pour le calcul de la transformée.

HLIN608 – Algorithmique du texte.

Alban MANCHERON

Alignement visuel de séquences

Recherche de motif

Extraction de motif

Contexte

Quelques programmes [presque] naïfs

Algorithmme KMR

Structures d'index

Hachage

Arbres

Automates

Tables

Extraction de motif

Structures d'index [Tables]

Définition (Table des suffixes)

Transformée de BURROWS-WHEELER du mot
ROUDOUDOU\$

R O U D O U D O U \$ +
\$ R O U D O U D O U
U \$ R O U D O U D O
O U \$ R O U D O U D
D O U \$ R O U D O U
U D O U \$ R O U D O
O U D O U \$ R O U D
D O U D O U \$ R O U
U D O U D O U \$ R O

Définition (FM-index)

Le FM-index (Ferragina–Manzini index) est la structure résultante de la compression de la transformée de BURROWS-WHEELER.

Sa construction est basée sur la table des suffixes pour le calcul de la transformée.

Extraction de motif

Structures d'index [Tables]

Définition (Table des suffixes)

Transformée de BURROWS-WHEELER du mot
ROUDOUDOU\$

R O U D O U D O U \$ +
\$ R O U D O U D O U
U \$ R O U D O U D O
O U \$ R O U D O U D
D O U \$ R O U D O U
U D O U \$ R O U D O
O U D O U \$ R O U D
D O U D O U \$ R O U
U D O U D O U \$ R O
O U D O U D O U \$ R

Le FM-index (Ferragina–Manzini index) est la structure résultante de la compression de la transformée de BURROWS-WHEELER. Sa construction est basée sur la table des suffixes pour le calcul de la transformée.

Extraction de motif

Structures d'index [Tables]

Définition (Table des suffixes)

Transformée de BURROWS-WHEELER du mot
ROUDOUDOU\$

R O U D O U D O U \$ +
\$ R O U D O U D O U
U \$ R O U D O U D O
O U \$ R O U D O U D tri commun entre deux suffixes
D O U \$ R O U D O U →
U D O U \$ R O U D O
O U D O U \$ R O U D $\mathcal{O}(n)$
D O U D O U \$ R O U
U D O U D O U \$ R O
O U D O U D O U \$ R

Le FM-index (Ferragina–Manzini index) est la structure résultante de la compression de la transformée de BURROWS-WHEELER. Sa construction est basée sur la table des suffixes pour le calcul de la transformée.

Extraction de motif

Structures d'index [Tables]

Définition (Table des suffixes)

Transformée de BURROWS-WHEELER du mot
ROUDOUDOU\$

R O U D O U D O U \$ + \$ R O U D O U D O U
\$ R O U D O U D O U
U \$ R O U D O U D O (longueurs de préfixes communs)
O U \$ R O U D O U D tri
D O U \$ R O U D O U commun entre deux suffixes
U D O U \$ R O U D O ↗
O U D O U \$ R O U D $\mathcal{O}(n)$
D O U D O U \$ R O U
U D O U D O U \$ R O
O U D O U D O U \$ R

Le FM-index (Ferragina–Manzini index) est la structure résultante de la compression de la transformée de BURROWS-WHEELER. Sa construction est basée sur la table des suffixes pour le calcul de la transformée.

Extraction de motif

Structures d'index [Tables]

Définition (Table des suffixes)

Transformée de BURROWS-WHEELER du mot
ROUDOUDOU\$

R O U D O U D O U \$ + \$ R O U D O U D O U
\$ R O U D O U D O U DOU \$ R O U D O U
U \$ R O U D O U D O tri (longueurs des préfixes communs)
O U \$ R O U D O U D commun entre deux suffixes
D O U \$ R O U D O U tri
U D O U \$ R O U D O \rightsquigarrow
O U D O U \$ R O U D $\mathcal{O}(n)$
D O U D O U \$ R O U
U D O U D O U \$ R O
O U D O U D O U \$ R

Le FM-index (Ferragina–Manzini index) est la structure résultante de la compression de la transformée de BURROWS-WHEELER. Sa construction est basée sur la table des suffixes pour le calcul de la transformée.

Extraction de motif

Structures d'index [Tables]

Définition (Table des suffixes)

Transformée de BURROWS-WHEELER du mot
ROUDOUDOU\$

R O U D O U D O U \$ \$ R O U D O U D O U U \$ R O U D O U D O O U \$ R O U D O U D D O U \$ R O U D O U U D O U \$ R O U D O O U D O U \$ R O U D D O U D O U \$ R O U U D O U D O U \$ R O O U D O U D O U \$ R	+ \$ R O U D O U D O U D O U \$ R O U D O U D O U D O U \$ R O U O U D O U \$ R O U D D O U D O U \$ R O U U D O U D O U \$ R O O U D O U D O U \$ R
--	--

tri

$\mathcal{O}(n)$

Le FM-index (Ferragina–Manzini index) est la structure résultante de la compression de la transformée de BURROWS-WHEELER. Sa construction est basée sur la table des suffixes pour le calcul de la transformée.

Extraction de motif

Structures d'index [Tables]

Définition (Table des suffixes)

Transformée de BURROWS-WHEELER du mot
ROUDOUDOU\$

R O U D O U D O U \$ \$ R O U D O U D O U U \$ R O U D O U D O O U \$ R O U D O U D D O U \$ R O U D O U U D O U \$ R O U D O O U D O U \$ R O U D D O U D O U \$ R O U U D O U D O U \$ R O U D O U D O U \$ R	+ tri 	\$ R O U D O U D O U D O U \$ R O U D O U D O U D O U \$ R O U O U \$ R O U D O U D O U \$ R O U D O U D D O U D O U \$ R O U U D O U D O U \$ R
---	---	---

Le FM-index (Ferragina–Manzini index) est la structure résultante de la compression de la transformée de BURROWS-WHEELER. Sa construction est basée sur la table des suffixes pour le calcul de la transformée.

Extraction de motif

Structures d'index [Tables]

Définition (Table des suffixes)

Transformée de BURROWS-WHEELER du mot
ROUDOUDOU\$

R O U D O U D O U \$ \$ R O U D O U D O U U \$ R O U D O U D O O U \$ R O U D O U D D O U \$ R O U D O U U D O U \$ R O U D O O U D O U \$ R O U D D O U D O U \$ R O U U D O U D O U \$ R O U D O U D O U \$ R	+ tri 	\$ R O U D O U D O U D O U \$ R O U D O U D O U D O U \$ R O U O U \$ R O U D O U D O U D O U \$ R O U D O U D O U \$ R O U
---	---	--

Le FM-index (Ferragina–Manzini index) est la structure résultante de la compression de la transformée de BURROWS-WHEELER. Sa construction est basée sur la table des suffixes pour le calcul de la transformée.

Extraction de motif

Structures d'index [Tables]

Définition (Table des suffixes)

Transformée de BURROWS-WHEELER du mot
ROUDOUDOU\$

R O U D O U D O U \$ \$ R O U D O U D O U U \$ R O U D O U D O O U \$ R O U D O U D D O U \$ R O U D O U C U D O U \$ R O U D O I O U D O U \$ R O U D D O U D O U \$ O U D O U D O I O U D O U D D O U D O U O U D O U I O U D O D O U O U	+ \$ R O U D O U D O U D O U \$ R O U D O U D O U D O U \$ R O U O U \$ R O U D O U D O U D O U \$ R O U D O U D O U D O U \$ R O U D O U D O U \$ R O U D O U D O U D O U \$ R O O U D O U D O U \$ R
---	--

tri  $\mathcal{O}(n)$

Le FM-index (Ferragina–Manzini index) est la structure résultante de la compression de la transformée de BURROWS-WHEELER. Sa construction est basée sur la table des suffixes pour le calcul de la transformée.

Extraction de motif

Structures d'index [Tables]

Définition (Table des suffixes)

Transformée de BURROWS-WHEELER du mot
ROUDOUDOU\$

R O U D O U D O U \$ \$ R O U D O U D O U U \$ R O U D O U D O O U \$ R O U D O U D D O U \$ R O U D O U C U D O U \$ R O U D O I D E F U D O U D E F I U D O U D E F I D U D O D E F I D E U D E F I D E F	+ tri	\$ R O U D O U D O U D O U \$ R O U D O U D O U D O U \$ R O U O U \$ R O U D O U D O U D O U \$ R O U D O U D O U D O U \$ R $\mathcal{O}(n)$ R O U D O U D O U \$ D O U D O U \$ R O U U D O U D O U \$ R O O U D O U D O U \$ R
---	--	---

Le FM-index (Ferragina–Manzini index) est la structure résultante de la compression de la transformée de BURROWS-WHEELER. Sa construction est basée sur la table des suffixes pour le calcul de la transformée.

Extraction de motif

Structures d'index [Tables]

Définition (Table des suffixes)

Transformée de BURROWS-WHEELER du mot
ROUDOUDOU\$

R O U D O U D O U \$ \$ R O U D O U D O U U \$ R O U D O U D O O U \$ R O U D O U D D O U \$ R O U D O U U D O U \$ R O U D O O U D O U \$ R O U D D O U D O U \$ R O U U D O U D O U \$ R	+ tri 	\$ R O U D O U D O U D O U \$ R O U D O U D O U D O U \$ R O U O U \$ R O U D O U D O U D O U \$ R O U D O U D O U D O U \$ R U \$ R O U D O U D O
--	--	---

Le FM-index (Ferragina–Manzini index) est la structure résultante de la compression de la transformée de BURROWS-WHEELER. Sa construction est basée sur la table des suffixes pour le calcul de la transformée.

Extraction de motif

Structures d'index [Tables]

Définition (Table des suffixes)

Transformée de BURROWS-WHEELER du mot
ROUDOUDOU\$

R	O	U	D	O	U	D	O	U	\$	+	\$	R	O	U	D	O	U	D	O	U
\$	R	O	U	D	O	U	D	O	U		DO	U	\$	R	O	U	D	O	U	
U	\$	R	O	U	D	O	U	D	O		DO	U	D	O	U	\$	R	O	U	
O	U	\$	R	O	U	D	O	U	D		OU	\$	R	O	U	D	O	U	D	
D	O	U	\$	R	O	U	D	O	U	tri	O	U	D	O	U	\$	R	O	U	
U	D	O	U	\$	R	O	U	D	O		OU	D	O	U	D	O	U	\$	R	
O	U	D	O	U	\$	R	O	U	D	$\mathcal{O}(n)$	R	O	U	D	O	U	D	O	U	
D	O	U	D	O	U	\$	R	O	U		OU	D	O	U	D	O	U	D	O	
U	D	O	U	D	O	U	\$	R	O		UD	O	U	D	O	\$	R	O	U	
O	U	D	O	U	D	O	U	\$	R		UD	O	U	D	O	U	\$	R	O	

Le FM-index (Ferragina–Manzini index) est la structure résultante de la compression de la transformée de BURROWS-WHEELER. Sa construction est basée sur la table des suffixes pour le calcul de la transformée.

Extraction de motif

Structures d'index [Tables]



Définition (Table des suffixes)

Transformée de BURROWS-WHEELER du mot ROUDOUDOU\$

R	O	U	D	O	U	D	O	U	\$		\$	R	O	U	D	O	U	D	O	U	
\$	R	O	U	D	O	U	D	O	U			DOU\$	R	O	U	D	O	U			
U	\$	R	O	U	D	O	U	D	O			DOUDOU\$	R	O	U						
O	U	\$	R	O	U	D	O	U	D			OU\$	R	O	U	D	O	U			
D	O	U	\$	R	O	U	D	O	U			DOUDOU\$	R	O	U	D	O	U			
U	D	O	U	\$	R	O	U	D	O			OU DOUDOU\$	R	O	U	D	O	U			
O	U	D	O	U	\$	R	O	U	D			O(n)	R	O	U	D	O	U			
D	O	U	D	O	U	\$	R	O	U				U\$	R	O	U	D	O	U		
U	D	O	U	D	O	U	\$	R	O				UDOUDOU\$	R	O	U	D	O	U		
O	U	D	O	U	D	O	U	\$	R				UDOUDOU\$	R	O	U	D	O	U		

Le FM-index (Ferragina–Manzini index) est la structure résultante de la compression de la transformée de BURROWS-WHEELER. Sa construction est basée sur la table des suffixes pour le calcul de la transformée.

Alignement visuel de séquences

Recherche de motif

Extraction de motif

Automates

Extraction de motif

Structures d'index [Tables]

Définition (Table des suffixes)

Transformée de BURROWS-WHEELER du mot
ROUDOUDOU\$

R	O	U	D	O	U	D	O	U	\$
\$	R	O	U	D	O	U	D	O	U
U	\$	R	O	U	D	O	U	\$	R
O	U	\$	R	O	U	D	O	U	D
D	O	U	\$	R	O	U	D	O	U
U	D	O	U	\$	R	O	U	D	O
O	U	D	O	U	\$	R	O	U	D
D	O	U	D	O	U	\$	R	O	U
U	D	O	U	D	O	\$	R	O	U
O	U	D	O	U	D	O	\$	R	O

+
tri

$\mathcal{O}(n)$

→

7

B	W	\simeq	S	A
T				
U				
U				
U				
D				
D				
D				
D				
O				
O				
O				
O				
O				

Le FM-index (Ferragina–Manzini index) est la structure résultante de la compression de la transformée de BURROWS-WHEELER. Sa construction est basée sur la table des suffixes pour le calcul de la transformée.

Extraction de motif

Structures d'index [Tables]

Définition (Table des suffixes)

Transformée de BURROWS-WHEELER du mot
ROUDOUDOU\$

+

Définition (Table des plus long préfixes communs)

longueur du plus long préfixe commun entre deux suffixes consécutifs de la table des suffixes.

Amélioration des performances de recherche

Définition (FM-index)

Le FM-index (Ferragina–Manzini index) est la structure résultante de la compression de la transformée de BURROWS-WHEELER. Sa construction est basée sur la table des suffixes pour le calcul de la transformée.



Extraction de motif

Structures d'index [Tables]

Définition (Table des suffixes)

Transformée de BURROWS-WHEELER du mot
ROUDOUDOU\$

Texte d'origine R₀ O₁ U₂ D₃ O₄ U₅ D₆ O₇ U₈ \$₉

Definition (Table des suffixes communs)

BWT	U ₈ U ₅ U ₂ D ₆ D ₃ R ₀ \$ ₉ O ₇ O ₄ O ₁
SA	\$ ₉ D ₆ D ₃ O ₇ O ₄ O ₁ R ₀ U ₈ U ₅ U ₂

Amélioration des performances de recherche

Definition (FM-index)

Le FM-index (Ferragina–Manzini index) est la structure résultante de la compression de la transformée de BURROWS-WHEELER.

Sa construction est basée sur la table des suffixes pour le calcul de la transformée.

Extraction de motif

Structures d'index [Tables]

Définition (Table des suffixes)

Transformée de BURROWS-WHEELER du mot
ROUDOUDOU\$

Texte d'origine R₀ O₁ U₂ D₃ O₄ U₅ D₆ O₇ U₈ \$₉

Definition (Table des suffixes communs)

BWT	U ₈ U ₅ U ₂ D ₆ D ₃ R ₀ \$ ₉ O ₇ O ₄ O ₁
SA	\$ ₉ D ₆ D ₃ O ₇ O ₄ O ₁ R ₀ U ₈ U ₅ U ₂

Amélioration des performances de recherche

Definition (FM-index)

Le FM-index (Ferragina–Manzini index) est la structure résultante de la compression de la transformée de BURROWS-WHEELER.

Sa construction est basée sur la table des suffixes pour le calcul de la transformée.

Extraction de motif

Structures d'index [Tables]

Définition (Table des suffixes)

Transformée de BURROWS-WHEELER du mot
ROUDOUDOU\$

Texte d'origine R₀ O₁ U₂ D₃ O₄ U₅ D₆ O₇ U₈ \$₉

Definition (Table des suffixes les plus courants)

BWT	U ₈ U ₅ U ₂ D ₆ D ₃ R ₀ \$ ₉ O ₇ O ₄ O ₁
SA	\$ ₉ D ₆ D ₃ O ₇ O ₄ O ₁ R ₀ U ₈ U ₅ U ₂

Amélioration des performances de recherche

Definition (FM-index)

Le FM-index (Ferragina–Manzini index) est la structure résultante

Recherche du mot UDOU

de la compression de la transformée de BURROWS-WHEELER.

Sa construction est basée sur la table des suffixes pour le calcul de la transformée.

Extraction de motif

Structures d'index [Tables]

Définition (Table des suffixes)

Transformée de BURROWS-WHEELER du mot
ROUDOUDOU\$

Texte d'origine R₀ O₁ U₂ D₃ O₄ U₅ D₆ O₇ U₈ \$₉

BWT U₈ U₅ U₂ D₆ D₃ R₀ \$₉ O₇ O₄ O₁
SA \$₉ D₆ D₃ O₇ O₄ O₁ R₀ U₈ U₅ U₂

Amélioration des performances de recherche

Définition (FM-index)

Le FM-index (Ferragina–Manzini index) est la structure résultante

Recherche du mot UDOU

de la compression de la transformée de BURROWS-WHEELER.

Sa construction est basée sur la table des suffixes pour le calcul de

la transformée.

Extraction de motif

Structures d'index [Tables]

Définition (Table des suffixes)

Transformée de BURROWS-WHEELER du mot
ROUDOUDOU\$

Texte d'origine R₀ O₁ U₂ D₃ O₄ U₅ D₆ O₇ U₈ \$₉

BWT U₈ U₅ U₂ D₆ D₃ R₀ \$₉ O₇ O₄ O₁
SA \$₉ D₆ D₃ O₇ O₄ O₁ R₀ U₈ U₅ U₂

U₈
U₅
U₂

Définition (FM-index)

Le FM-index (Ferragina–Manzini index) est la structure résultante

de la compression de la transformée de BURROWS-WHEELER.

Sa construction est basée sur la table des suffixes pour le calcul de la transformée.

Extraction de motif

Structures d'index [Tables]

Définition (Table des suffixes)

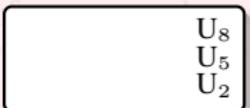
Transformée de BURROWS-WHEELER du mot
ROUDOUDOU\$

Texte d'origine R₀ O₁ U₂ D₃ O₄ U₅ D₆ O₇ U₈ \$₉

Inition (Table des suffixes communs)

BWT	U ₈ U ₅ U ₂ D ₆ D ₃ R ₀ \$ ₉ O ₇ O ₄ O ₁
SA	\$ ₉ D ₆ D ₃ O ₇ O ₄ O ₁ R ₀ U ₈ U ₅ U ₂

Amélioration des performances de recherche



Définition (FM-index)

Le FM-index (Ferragina–Manzini index) est la structure résultante

Recherche du mot UD**OU**

de la compression de la transformée de BURROWS-WHEELER.

Sa construction est basée sur la table des suffixes pour le calcul de la transformée.

Extraction de motif

Structures d'index [Tables]

Définition (Table des suffixes)

Transformée de BURROWS-WHEELER du mot
ROUDOUDOU\$

Texte d'origine R₀ O₁ U₂ D₃ O₄ U₅ D₆ O₇ U₈ \$₉

Definition (Table des suffixes communs)

BWT	U ₈ U ₅ U ₂ D ₆ D ₃ R ₀ \$ ₉ O ₇ O ₄ O ₁
SA	\$ ₉ D ₆ D ₃ O ₇ O ₄ O ₁ R ₀ U ₈ U ₅ U ₂

Amélioration des performances de recherche



Definition (FM-index)

Le FM-index (Ferragina–Manzini index) est la structure résultante

de la compression de la transformée de BURROWS-WHEELER.

Sa construction est basée sur la table des suffixes pour le calcul de la transformée.

Extraction de motif

Structures d'index [Tables]

Définition (Table des suffixes)

Transformée de BURROWS-WHEELER du mot
ROUDOUDOU\$

Texte d'origine R₀ O₁ U₂ D₃ O₄ U₅ D₆ O₇ U₈ \$₉

Inition (Table des suffixes communs)

BWT	U ₈ U ₅ U ₂ D ₆ D ₃ R ₀ \$ ₉ O ₇ O ₄ O ₁
SA	\$ ₉ D ₆ D ₃ O ₇ O ₄ O ₁ R ₀ U ₈ U ₅ U ₂

Amélioration des performances de recherche

O ₇	U ₈
O ₄	U ₅

Définition (FM-index)

Le FM-index (Ferragina–Manzini index) est la structure résultante

Recherche du mot UD**OU**

de la compression de la transformée de BURROWS-WHEELER.

Sa construction est basée sur la table des suffixes pour le calcul de la transformée.

Extraction de motif

Structures d'index [Tables]

Définition (Table des suffixes)

Transformée de BURROWS-WHEELER du mot
ROUDOUDOU\$

Texte d'origine R₀ O₁ U₂ D₃ O₄ U₅ D₆ O₇ U₈ \$₉

Definition (Table des suffixes communs)

BWT	U ₈ U ₅ U ₂ D ₆ D ₃ R ₀ \$ ₉ O ₇ O ₄ O ₁
SA	\$ ₉ D ₆ D ₃ O ₇ O ₄ O ₁ R ₀ U ₈ U ₅ U ₂

Amélioration des performances de recherche

O ₇	U ₈
O ₄	U ₅

Definition (FM-index)

Le FM-index (Ferragina–Manzini index) est la structure résultante

Recherche du mot UDOU

de la compression de la transformée de BURROWS-WHEELER.

Sa construction est basée sur la table des suffixes pour le calcul de la transformée.

Extraction de motif

Structures d'index [Tables]

Définition (Table des suffixes)

Transformée de BURROWS-WHEELER du mot
ROUDOUDOU\$

Texte d'origine R₀ O₁ U₂ D₃ O₄ U₅ D₆ O₇ U₈ \$₉

Definition (Table des suffixes les plus courants)

BWT	U ₈ U ₅ U ₂ D ₆ D ₃ R ₀ \$ ₉ O ₇ O ₄ O ₁
SA	\$ ₉ D ₆ D ₃ O ₇ O ₄ O ₁ R ₀ U ₈ U ₅ U ₂

Amélioration des performances de recherche

O ₇	U ₈
O ₄	U ₅

Definition (FM-index)

Le FM-index (Ferragina–Manzini index) est la structure résultante

Recherche du mot UDOU

de la compression de la transformée de BURROWS-WHEELER.

Sa construction est basée sur la table des suffixes pour le calcul de la transformée.

Extraction de motif

Structures d'index [Tables]

Définition (Table des suffixes)

Transformée de BURROWS-WHEELER du mot
ROUDOUDOU\$

Texte d'origine R₀ O₁ U₂ D₃ O₄ U₅ D₆ O₇ U₈ \$₉

Définition (Table des suffixes les plus courants)

BWT	U ₈ U ₅ U ₂ D ₆ D ₃ R ₀ \$ ₉ O ₇ O ₄ O ₁
SA	\$ ₉ D ₆ D ₃ O ₇ O ₄ O ₁ R ₀ U ₈ U ₅ U ₂

Amélioration des performances de recherche

D ₆	O ₇	U ₈
D ₃	O ₄	U ₅

Définition (FM-index)

Le FM-index (Ferragina–Manzini index) est la structure résultante

Recherche du mot UDOU

de la compression de la transformée de BURROWS-WHEELER.

Sa construction est basée sur la table des suffixes pour le calcul de la transformée.

Extraction de motif

Structures d'index [Tables]

Définition (Table des suffixes)

Transformée de BURROWS-WHEELER du mot
ROUDOUDOU\$

Texte d'origine R₀ O₁ U₂ D₃ O₄ U₅ D₆ O₇ U₈ \$₉

Definition (Table des suffixes communs)

BWT	U ₈ U ₅ U ₂ D ₆ D ₃ R ₀ \$ ₉ O ₇ O ₄ O ₁
SA	\$ ₉ D ₆ D ₃ O ₇ O ₄ O ₁ R ₀ U ₈ U ₅ U ₂

Amélioration des performances de recherche

D ₆	O ₇	U ₈
D ₃	O ₄	U ₅

Definition (FM-index)

Le FM-index (Ferragina–Manzini index) est la structure résultante

Recherche du mot **UDOU**

de la compression de la transformée de BURROWS-WHEELER.

Sa construction est basée sur la table des suffixes pour le calcul de la transformée.

Extraction de motif

Structures d'index [Tables]

Définition (Table des suffixes)

Transformée de BURROWS-WHEELER du mot
ROUDOUDOU\$

Texte d'origine R₀ O₁ U₂ D₃ O₄ U₅ D₆ O₇ U₈ \$₉

Definition (Table des suffixes communs)

BWT	U ₈ U ₅ U ₂ D ₆ D ₃ R ₀ \$ ₉ O ₇ O ₄ O ₁
SA	\$ ₉ D ₆ D ₃ O ₇ O ₄ O ₁ R ₀ U ₈ U ₅ U ₂

Amélioration des performances de recherche

U ₅	D ₆	O ₇	U ₈
U ₂	D ₃	O ₄	U ₅

Definition (FM-index)

Le FM-index (Ferragina–Manzini index) est la structure résultante

Recherche du mot **UDOU**

de la compression de la transformée de BURROWS-WHEELER.

Sa construction est basée sur la table des suffixes pour le calcul de la transformée.

Extraction de motif

Structures d'index [Tables]

Définition (Table des suffixes)

Transformée de BURROWS-WHEELER du mot
ROUDOUDOU\$

Texte d'origine R₀ O₁ U₂ D₃ O₄ U₅ D₆ O₇ U₈ \$₉

Définition (Table des suffixes communs)

BWT	U ₈ U ₅ U ₂ D ₆ D ₃ R ₀ \$ ₉ O ₇ O ₄ O ₁
SA	\$ ₉ D ₆ D ₃ O ₇ O ₄ O ₁ R ₀ U ₈ U ₅ U ₂

Amélioration des performances de recherche

U ₅	D ₆	O ₇	U ₈
U ₂	D ₃	O ₄	U ₅

Définition (FM-index)

Le FM-index (Ferragina–Manzini index) est la structure résultante de la compression de la transformée de BURROWS-WHEELER.

Sa construction est basée sur la table des suffixes pour le calcul de la transformée.



That's All, Folks !

Merci de votre attention.

Vous pouvez rentrer chez vous.

C'est fini.

Au revoir.

Bon courage.

Ceci n'est pas un test ophtalmologique,

mais si vous arrivez à lire jusqu'au bout . . .

. . . c'est que vous avez du temps.

Profitez-en pour programmer alors !!!

Alignement visuel de séquences

Recherche de motif

Extraction de motif

Contexte

Quelques programmes [presque] naïfs

Algorithmes KMR

Structures d'index

Hachage

Arbres

Automates

Tables