Filtrage de séquences d'ADN pour la recherche de longues répétitions multiples

Thèse d'informatique

Pierre Peterlongo

Institut Gaspard-Monge, Université de Marne-la-Vallée

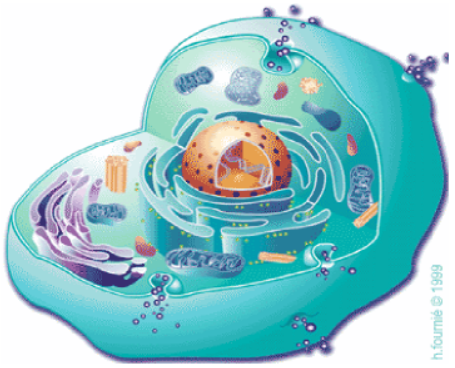
Marie-France Sagot
Maxime Crochemore

21 Septembre 2006

Plan de l'exposé

Plan de l'exposé

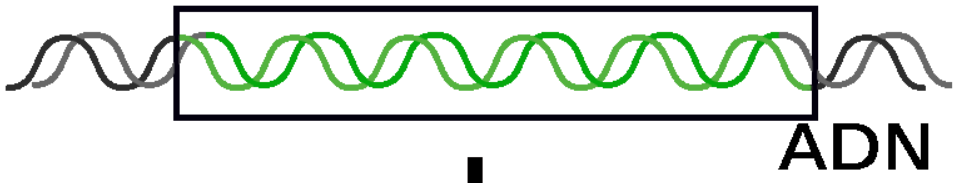
ADN et Synthèse protéique

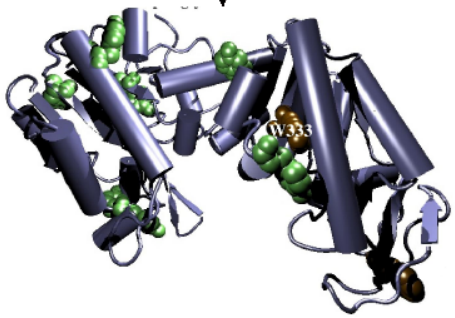


- Gène (Segment d'ADN codant pour une protéine)
- ARN
- Protéine (Constituant principal des cellules)

ADN et Synthèse protéique

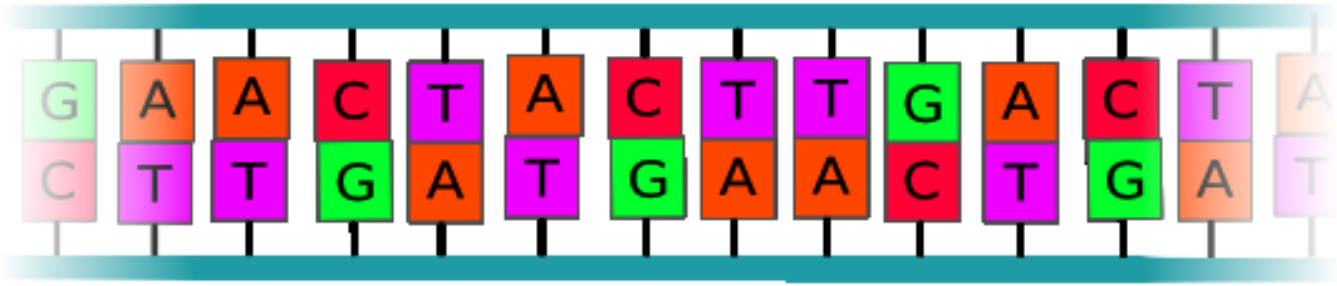
- Gène (Segment d'ADN codant pour une protéine)
- ARN
- Protéine (Constituant principal des cellules)



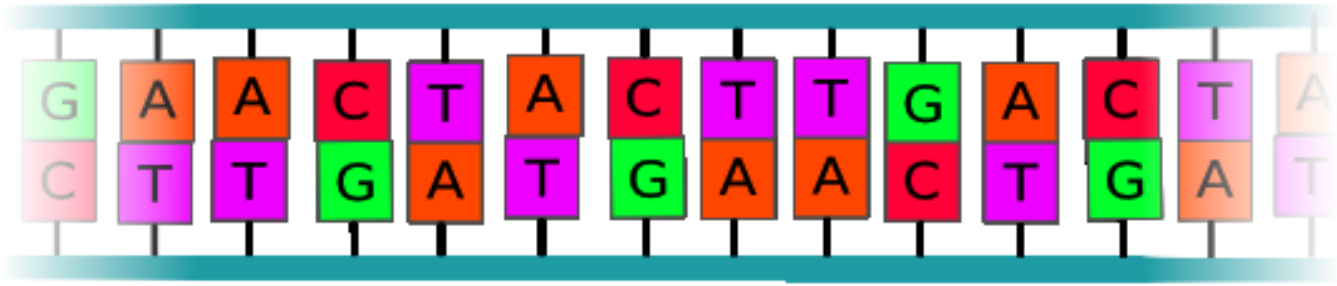


Protéine

Composition de l'ADN

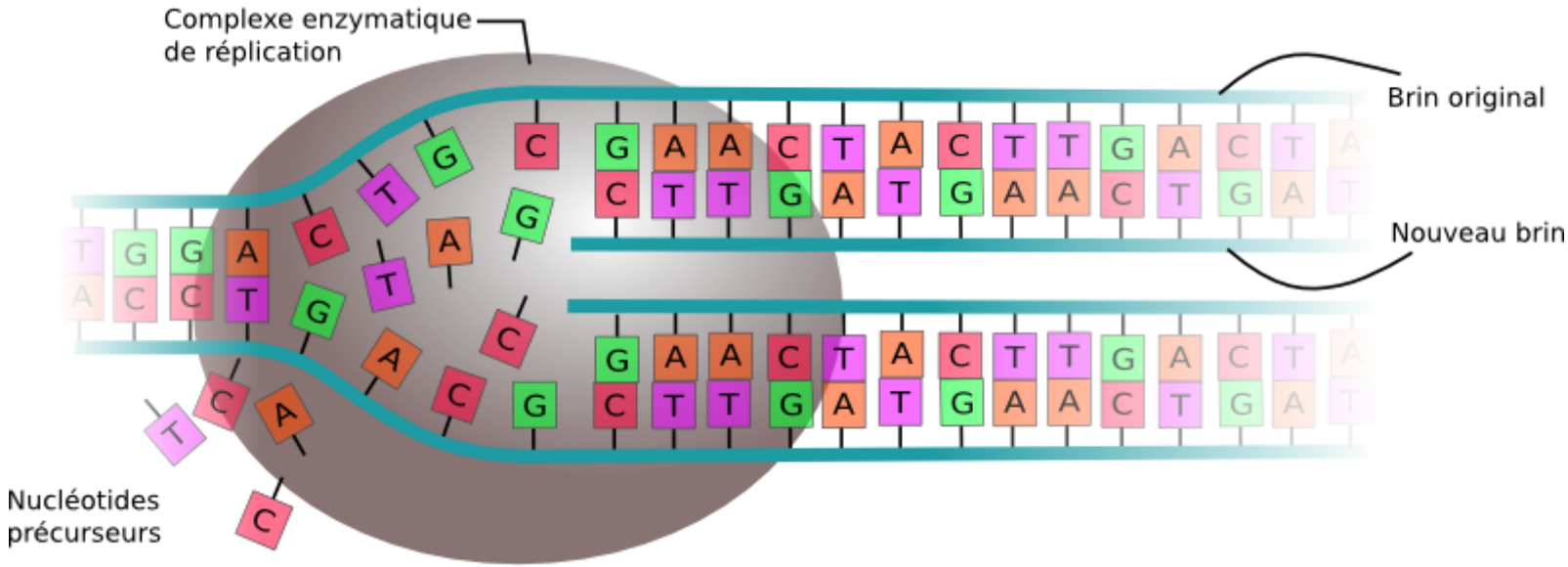


Composition de l'ADN



GAACTACTTGACTA

Réplication de l'ADN



- Réplication parfois imparfaite → création d'erreurs :
  - Insertion, suppression, substitution
- Pression de sélection
  - Portions codantes bien conservées
  - Portions non-codantes possiblement moins conservées

Similarités - Répétitions

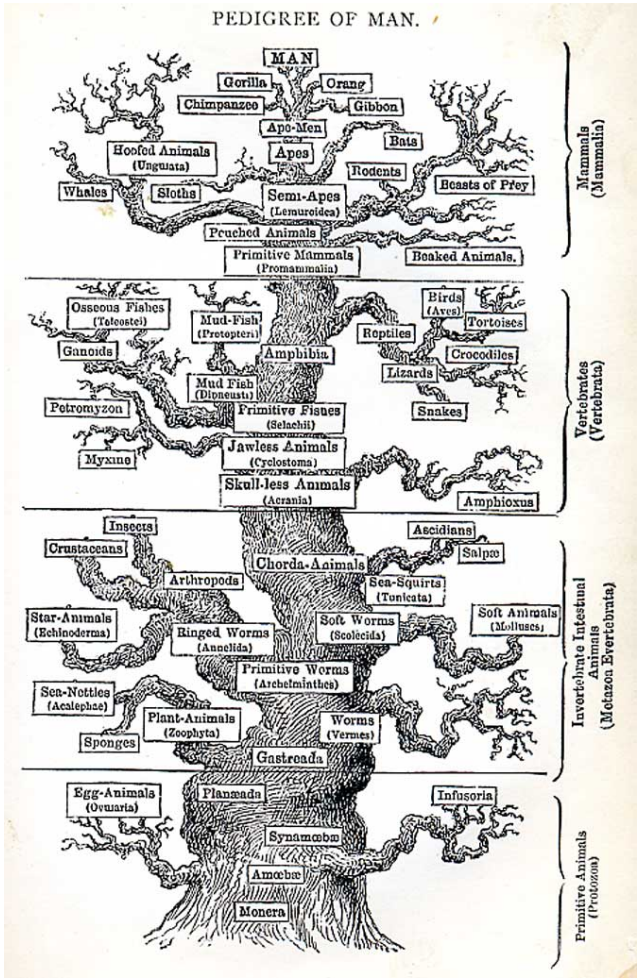
Utilisation

- Aide à l'annotation des génomes (transfert de connaissance)

Similarités - Répétitions

Utilisation

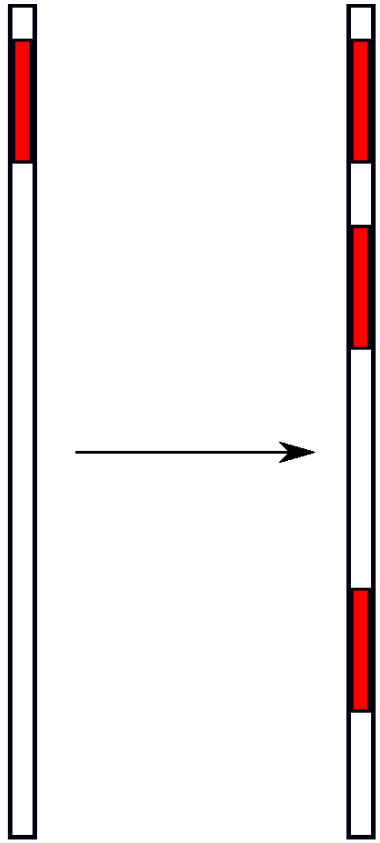
- Aide à l'annotation des génomes (transfert de connaissance)
- Phylogénie



Similarités - Répétitions

Répétitions sur les génomes - Éléments transposables

- Translocations (échanges de matériel génétique dans un génome)
- Transferts horizontaux (échanges entre espèces)



Conséquences

- Implication potentielle dans la régulation des gènes
- Rôle encore méconnu

Difficile à détecter

Notre but

Détecter des répétitions

- Longues : $>$ 100 paires de bases (pb)
- Dégénérées
- Multiples : nombre d'occurrences ≥ 2
- Dans de (très) grandes séquences : $>$ 1 Mb

Plan de l'exposé

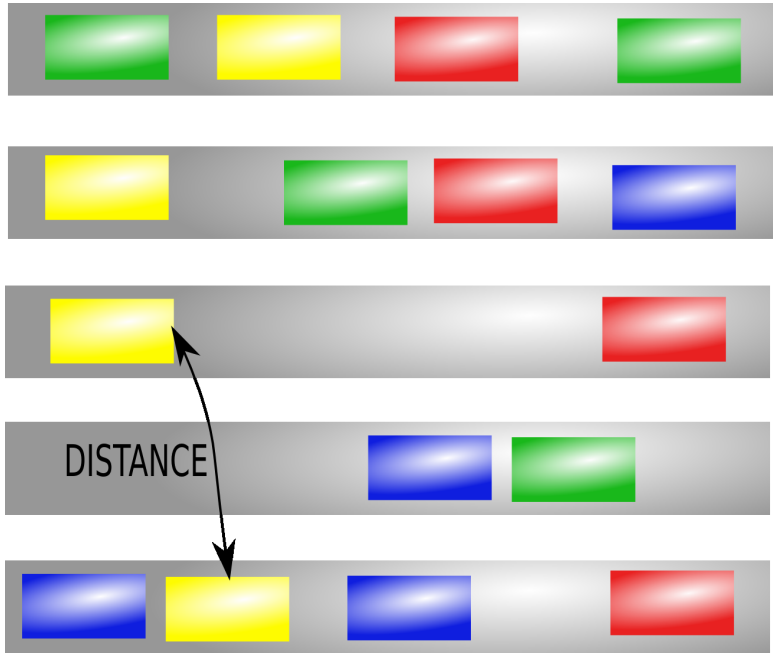
Motivations

- Extraction de longues répétitions approchées
- Distance :

Hamming : nombre minimum de substitutions

Édition : nombre minimum de substitutions, insertions, suppressions

ACTCGCA
ACTTGCA
A–TCGCA
ACTTG–A



Motivations

- Extraction de longues répétitions approchées
- Distance :

Hamming : nombre minimum de substitutions

Édition : nombre minimum de substitutions, insertions, suppressions

ACTCGCA ACTTGCA
A–TCGCA ACTTG–A

Solution existante

- $\approx$ Alignement multiple local
- Utilisation de la programmation dynamique

Motivations

Limites de l'alignement multiple

Complexité trop importante

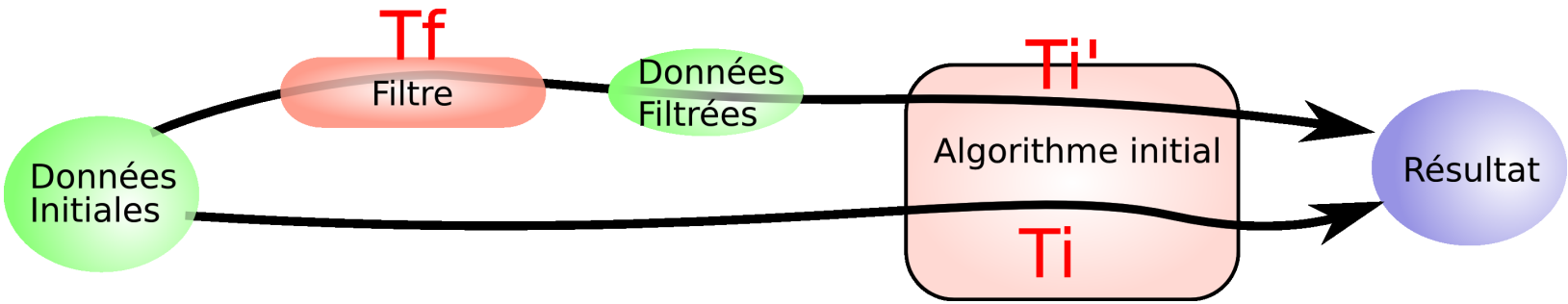
- Temps d'exécution
  $O(2^{\text{nombre de séquences}} \times \text{taille des séquences}^{\text{nombre de séquences}})$
- Trop long, par exemple :
  - Ordinateur *classique*, 10^9 opérations par secondes
  - 3 séquences de taille 1 mégabase : plusieurs années de calcul

- Il existe des solutions, mais ce sont des heuristiques

Solution proposée

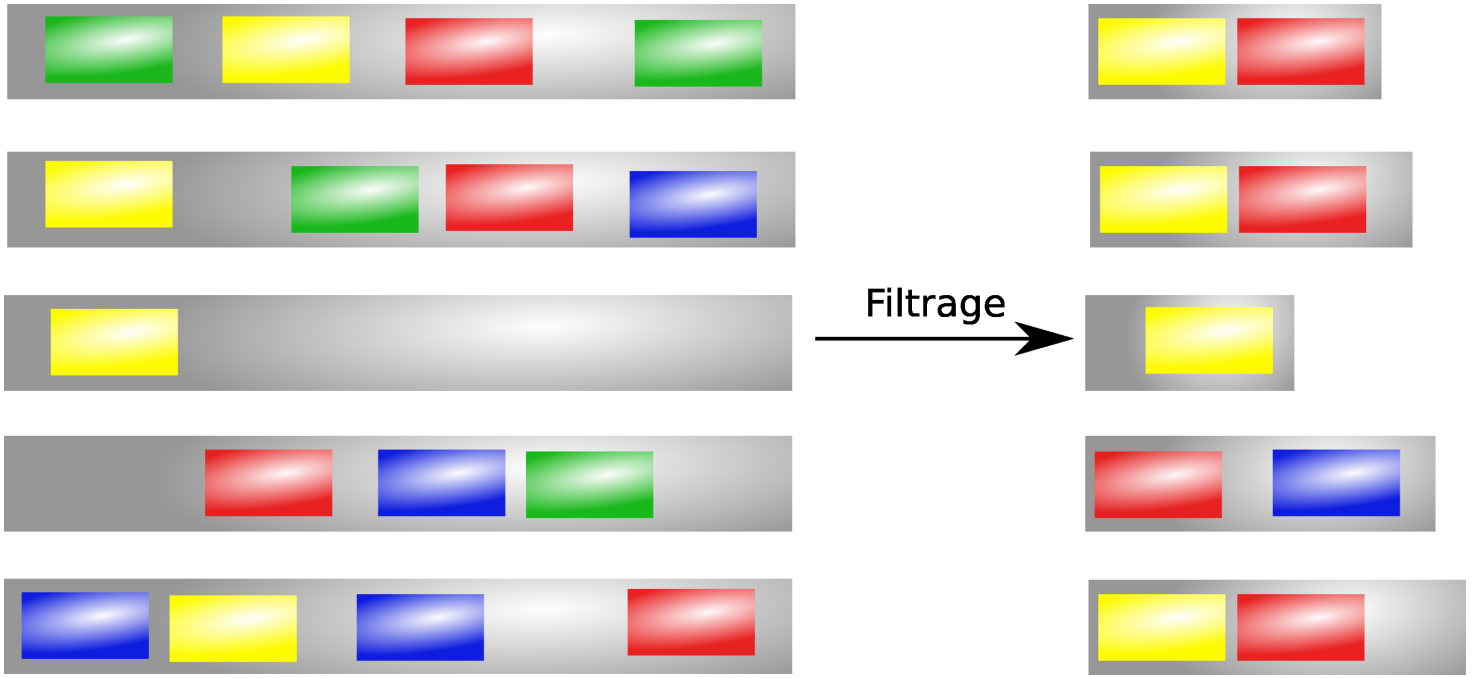
Filtre

- Suppression de données n'influant pas le résultat final (filtre exact)

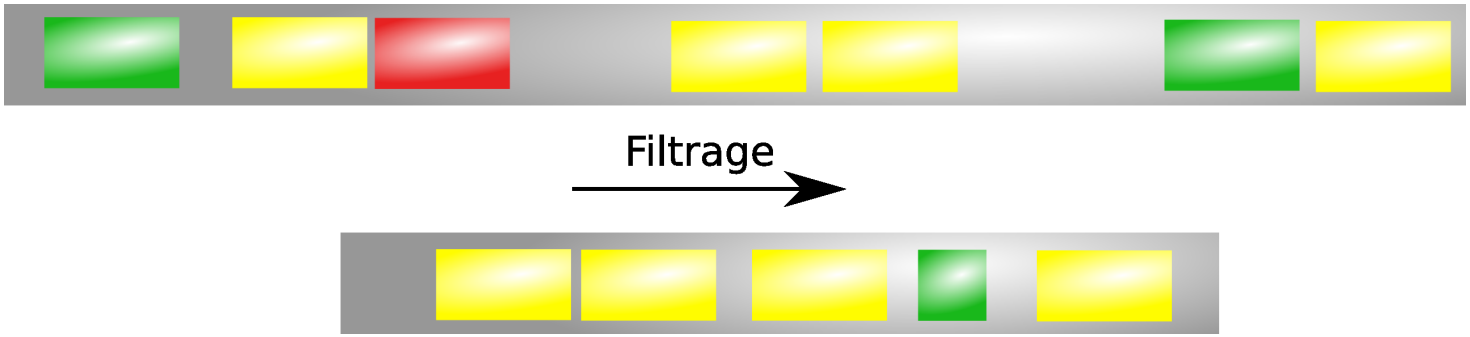


- $Tf + Ti' \ll Ti$

- Données - Filtrage - Alignement multiple local
- Suppression de portions ne respectant pas une condition nécessaire



- Données - Filtrage - Alignement multiple local
- Suppression de portions ne respectant pas une condition nécessaire



Autres algorithmes de filtrage

	Deux à deux	Multiple
Exact	Quasar[1], Swift[2]	$\emptyset$
Non exact	Yass[3]	Multi-lagan[4]

[1] Burkhardt, S. ; Crauser, A. ; Ferragina, P. ; Lenhof, H.P. ; Vingron, M. *q-Gram Based Database Searching Using a Suffix Array (QUASAR)*

[2] Rasmussen, K.R. ; Stoye, J. ; Myers, E.W. *Efficient q-Gram Filters for Finding All ϵ-Matches Over a Given Length*

[3] Noé, L. ; Kucherov, G. *YASS : enhancing the sensitivity of DNA similarity search*

[4] Brudno, M. ; Do, C.B. ; Cooper, G.M. ; Kim, M. ; Davydov, E. ; Green, E.D. ; Sidow, A. ; Batzoglou, S. *LAGAN and Multi-LAGAN : Efficient Tools for Large-Scale Multiple Alignment of Genomic DNA*

Idée de base des filtres de répétitions

- Entre deux mots *similaires*, des *sous-parties* sont exactement conservées.

ATTAAAATTT
ATAAATATTT

Idée de base des filtres de répétitions

- Entre deux mots *similaires*, des *sous-parties* sont exactement conservées.
- Si deux mots ne contiennent *pas assez* de *sous-parties* conservées, ils ne peuvent pas être *similaires*

ATTAAAATTT
ATAAATATTT

Plan de l'exposé

Nimbus - But précis

L'utilisateur donne

- L : Taille des répétitions
- r : Nombre d'occurrences minimum de chaque répétition
- d : Distance de Hamming maximale entre chaque paire de répétitions
- m séquences ($m \geq r$)

Nimbus donne :

- L'ensemble des positions conservées
- Les chaînes de caractères correspondantes

P. Peterlongo, N. Pisanti, F. Boyer, A. Pereira do Lago, M.-F. Sagot, *Lossless filter for multiple repetitions*, Journal of Discrete Algorithms, (en soumission)

P. Peterlongo, N. Pisanti, F. Boyer, M.-F. Sagot, *Lossless Filter for Finding Long Multiple Approximate Repetitions Using a New Data Structure, the Bi-Factor Array*, (SPIRE 2005)

Nombre de k-facteurs (non chevauchants ■■■) partagés

ATTAAAATTT
ATAAATATTT

Condition nécessaire - 2 mots (Pevzner 1995)

- Deux mots de taille L
- Distant d'au plus d **substitutions**
- Partagent au moins

$$\left\lfloor \frac{L}{k} \right\rfloor - d$$

k-facteurs non chevauchant

Nombre de k-facteurs (non chevauchants ▬▬▬) partagés

ATTAAAATTT
ATAAAATATTT
ATAAATATAT

Condition nécessaire - r mots

- r mots de taille L
- Distant d'au plus d **substitutions** 2 à 2
- Partagent au moins

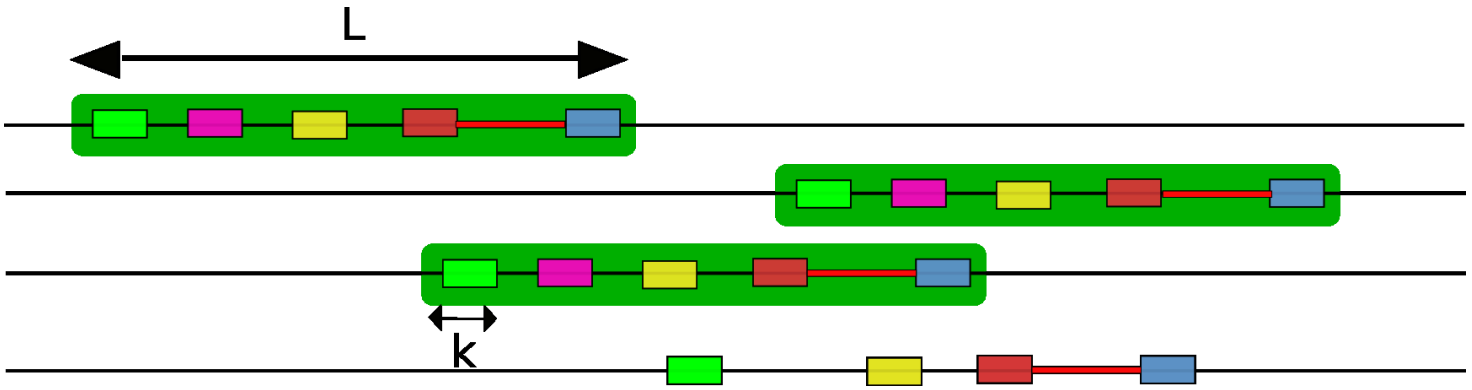
$$p = \left\lfloor \frac{L}{k} \right\rfloor - d - (r-2) \left\lfloor \frac{d}{2} \right\rfloor$$

k-facteurs non chevauchant

Application - Idée générale

Détection d'ensembles de zones

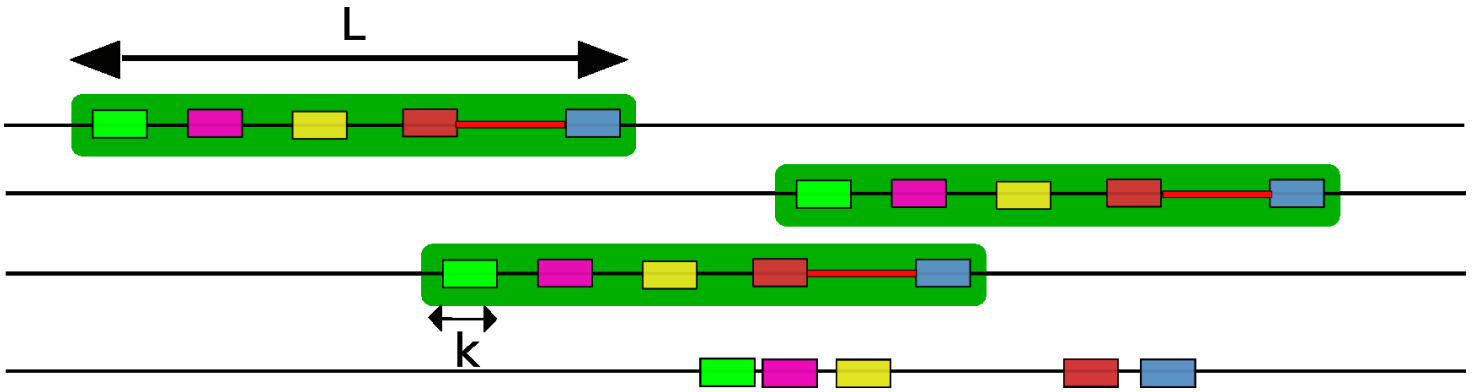
- De taille L
- Apparaissant au moins dans r séquences
- Partageant au moins $p(= 5)$ k-facteurs



Application - Idée générale

Détection d'ensembles de zones

- De taille L
- Apparaissant au moins dans r séquences
- Partageant au moins $p(=5)$ k-facteurs également répartis



Aparté - Motivation de **travaux annexes**

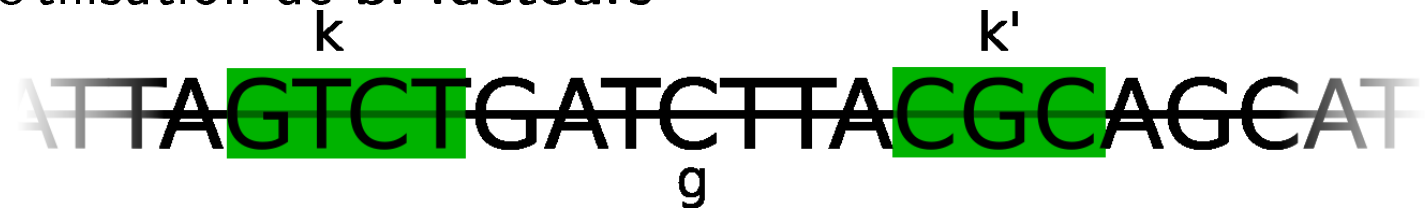
Indexation de **Motifs à Trous**

- Arbre des suffixes : C.S. Iliopoulos, J. McHugh, P. Peterlongo, N. Pisanti, W. Rytter, M.-F. Sagot, *A first approach to finding common motifs with gaps*, International Journal of Foundation of Computer Science
- Automate : P. Antoniou, J. Holub, C. S. Iliopoulos, B. Melichar, P. Peterlongo, *Finding Common Motifs with Gaps using Finite Automata*, (CIAA 2006)

Plan de l'exposé

Algorithme - Aperçu

Utilisation de **bi-facteurs**



Arbre des bi-facteurs

- P. Peterlongo, J. Allali, M.-F. Sagot, *The Gapped-Factor Tree*, PSC'06

Tableau des bi-facteurs

- Indexation en $O(n)$
- Liste des occurrences d'un bi-facteur en $O(1)$

Algorithme - Aperçu

Utilisation de **bi-facteurs**

Arbre des bi-facteurs

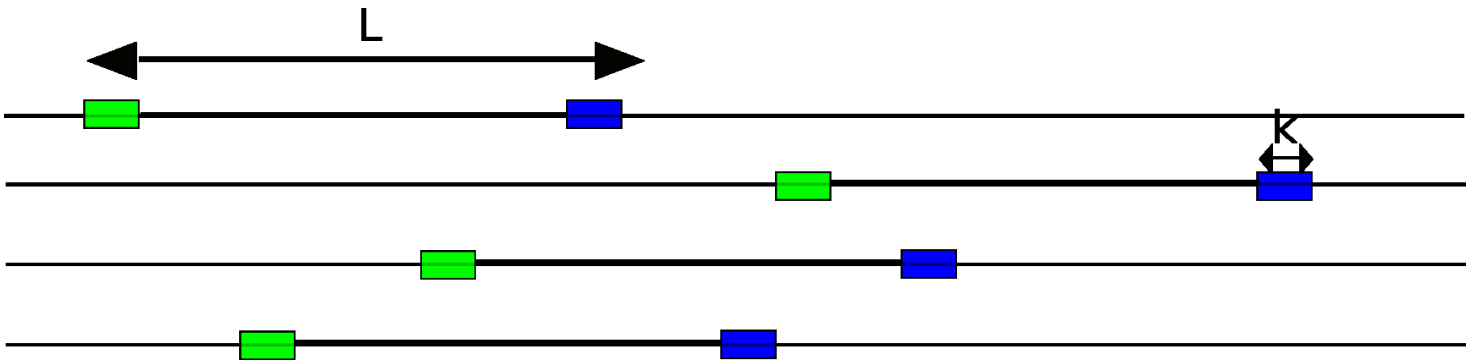
- P. Peterlongo, J. Allali, M.-F. Sagot, *The Gapped-Factor Tree*, PSC'06

Tableau des bi-facteurs

- Indexation en $O(n)$
- Liste des occurrences d'un bi-facteur en $O(1)$

Algorithme - Aperçu

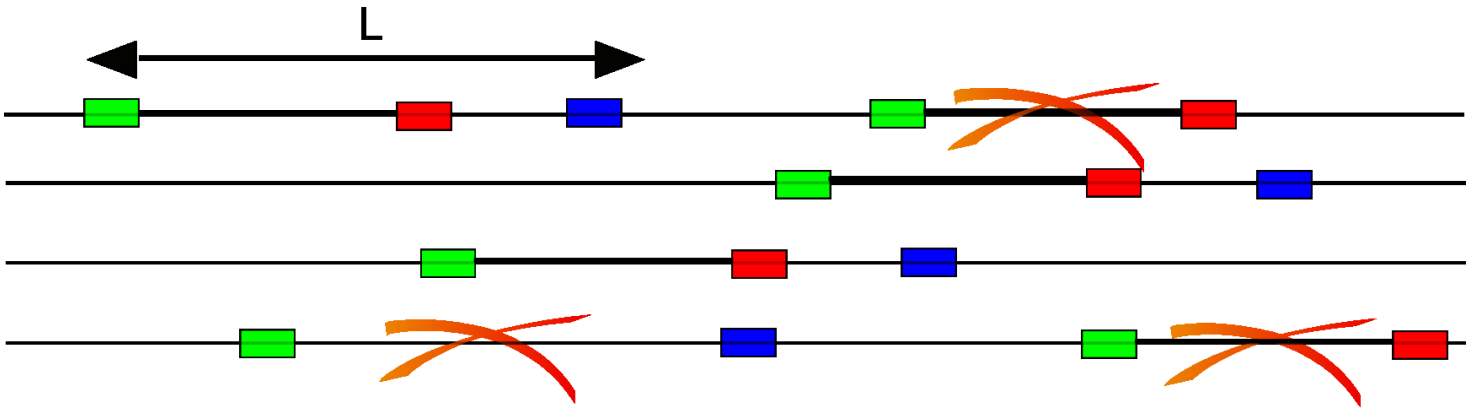
Détection d'un *grand*[5] bi-facteur apparaissant au moins r fois



[5] $(p-2) \times k \leq \text{trou} \leq L - 2k$

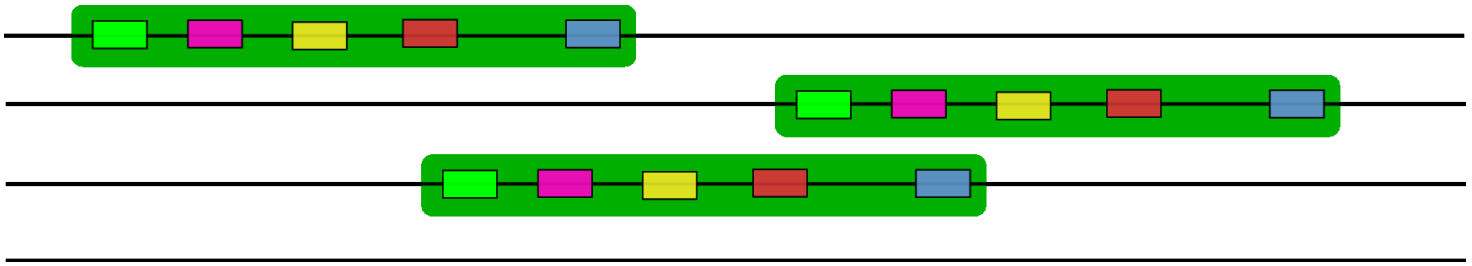
Algorithme - Aperçu

Ajout d'un bi-facteur ayant le même premier k-facteur



Algorithme - Aperçu

Répété jusqu'à obtenir p k-facteurs



Complexité

Temps

$$O(L \times n \times N \times Z^p)$$

Mémoire

$$O\left((L-2k) \times \frac{N}{|\Sigma|^k}\right)$$

- $Z = L \times \min(|\Sigma|^k, n)$
- n : Taille moyenne des séquences
- N : Taille totale des séquences ($N = n \times m$)
- p : Nombre minimum de k-facteurs
- $|\Sigma|$: Taille de l'alphabet (4)

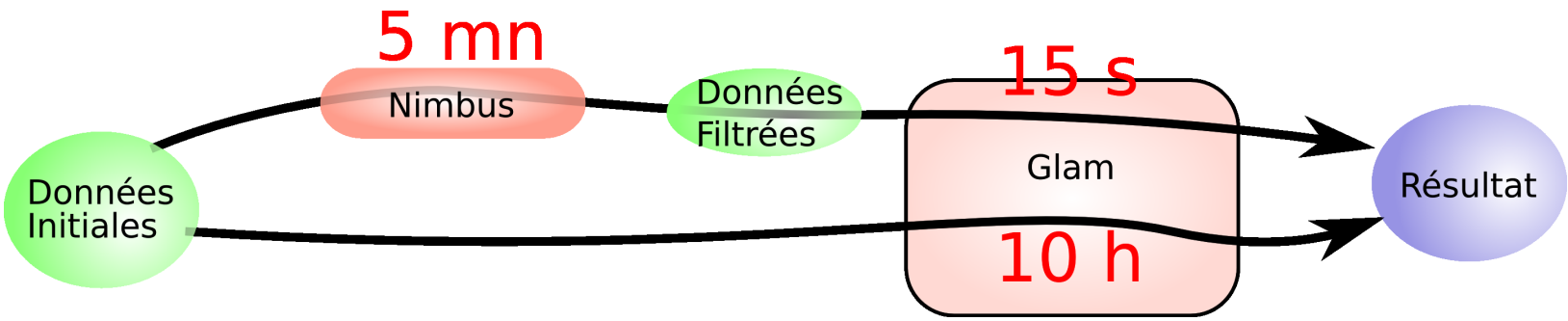
Plan de l'exposé

Accélération d'algorithmes d'alignements locaux

Pré-traitement de séquences (générées) pour accélérer Glam[5]

- 5 séqs. aléatoires,
- Taille totale 1 Mb,
- Répétitions 100, 10
- Paramètres : $L = 100, d = 10, r = 5, k = 5$

- Pentium III, 1200 MHz
- 512 Mo mémoire.



- $\approx$ 100 fois plus rapide

[5] Frith, M. ; Hansen, U. ; Spouge, J.L. & Weng, Z. *Finding Functionnal Sequence Elements by Multiple Local Alignement*

Limites de Nimbus

- Résultats limités (distance de Hamming)
- Inapplicable pour un grand nombre d'occurrences r

$$p = \left\lfloor \frac{L}{k} \right\rfloor - d - (r-2) \left\lfloor \frac{d}{2} \right\rfloor$$

Plan de l'exposé

Ed'Nimbus - But précis

L'utilisateur donne

- L : Taille des répétitions
- r : Nombre d'occurrences minimum de chaque répétition
- d : Distance d'édition maximale entre chaque paire de répétitions substitutions, insertions, suppressions
- m séquences ($m \geq r$)

Ed'Nimbus donne :

- L'ensemble des positions conservées
- Les chaînes de caractères correspondantes

P. Peterlongo, N. Pisanti, A. Pereira do Lago, M.-F. Sagot, *Ed'Nimbus : A Lossless Filter for Long Multiple Repetitions with Edit Distance* , Poster, Jobim 2006

P. Peterlongo, N. Pisanti, A. Pereira do Lago, M.-F. Sagot, *Lossless Filter for Long Multiple Repetitions with Edit Distance*, Rapport de recherche, Università di Pisa

Nombre de k-facteurs (possiblement chevauchants ) partagés

ATTAAAATTT
ATAAAATTT

Nombre de k-facteurs (possiblement chevauchants) partagés

ATTAAAATTT
ATAAAATTT

Condition nécessaire

- Deux mots de taille L
- Distant d'au plus d **opérations d'édition**
- Partagent au moins

$$p = L - (d+1) \times k + 1$$

k-facteurs

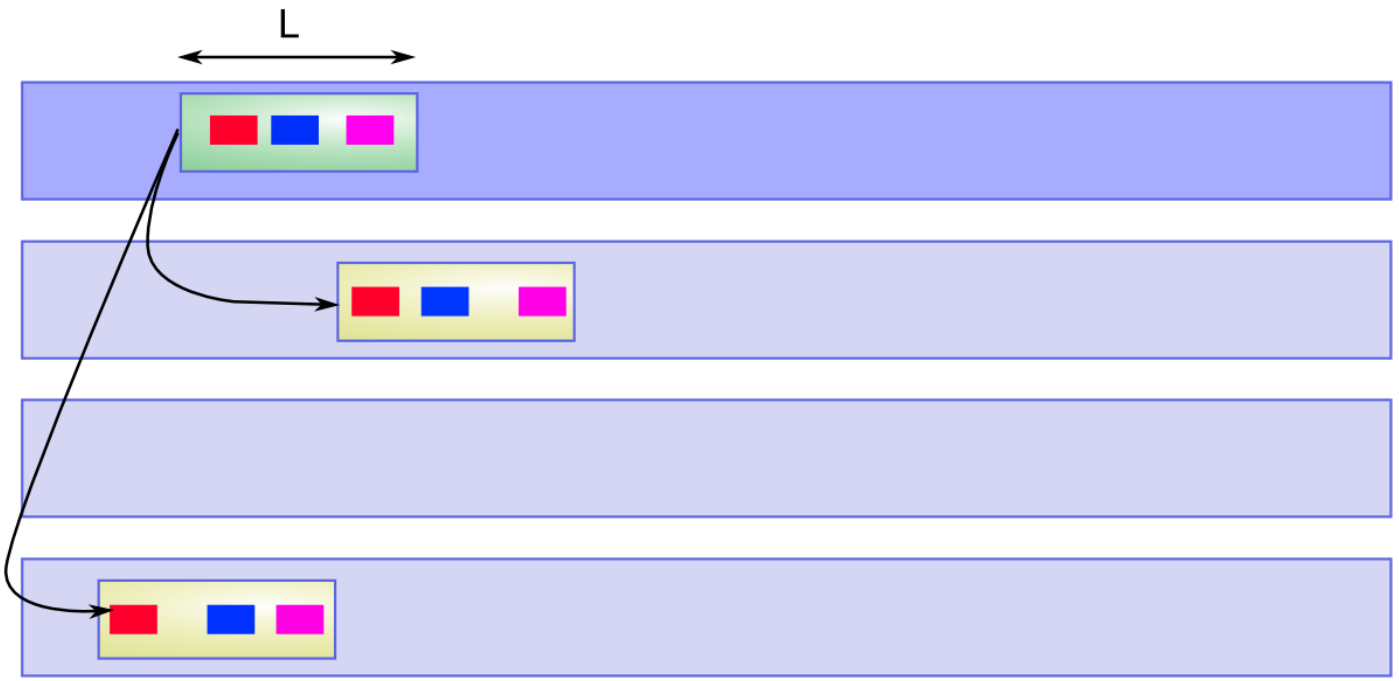
Application - Idée générale

Sur chaque séquence, une fenêtre **glisse** pour tester si elle peut être une répétition

Application - Idée générale

Toute paire respecte la condition sur p

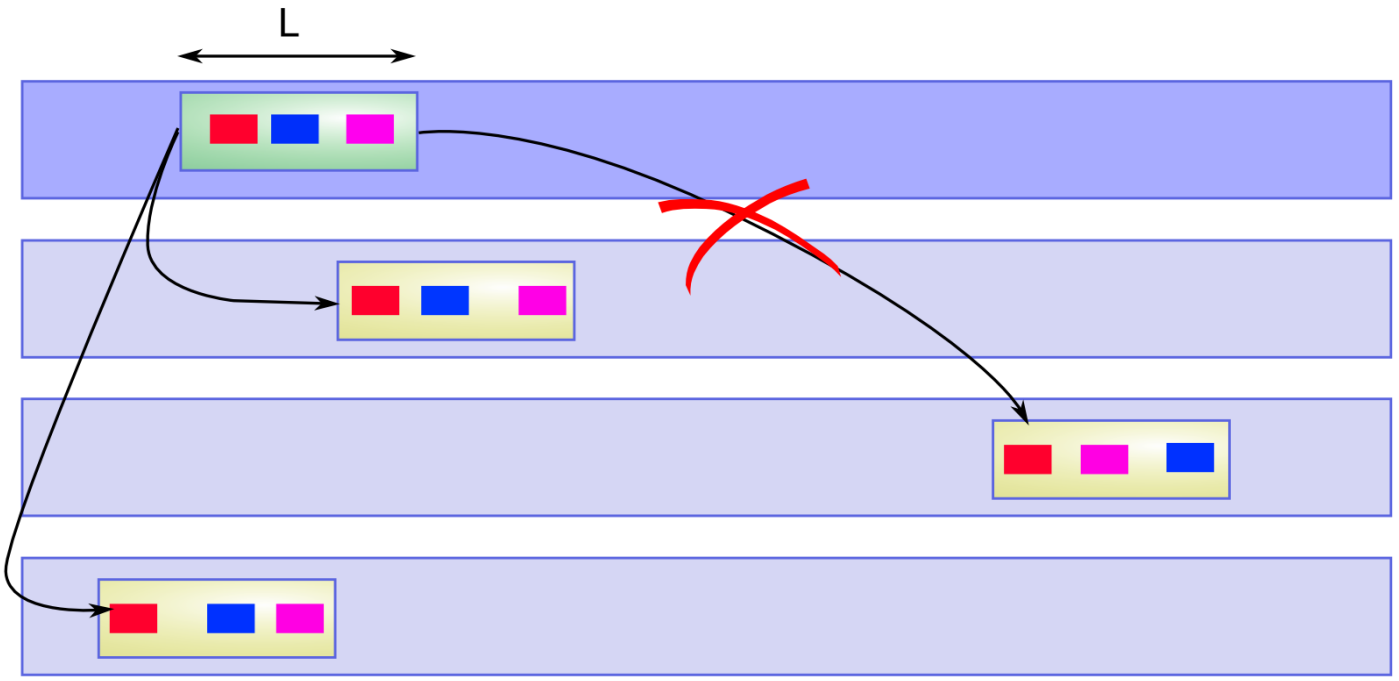
- $p = 3$ (nombre minimum de k-facteurs partagés)



Application - Idée générale

Toute paire respecte la condition sur p

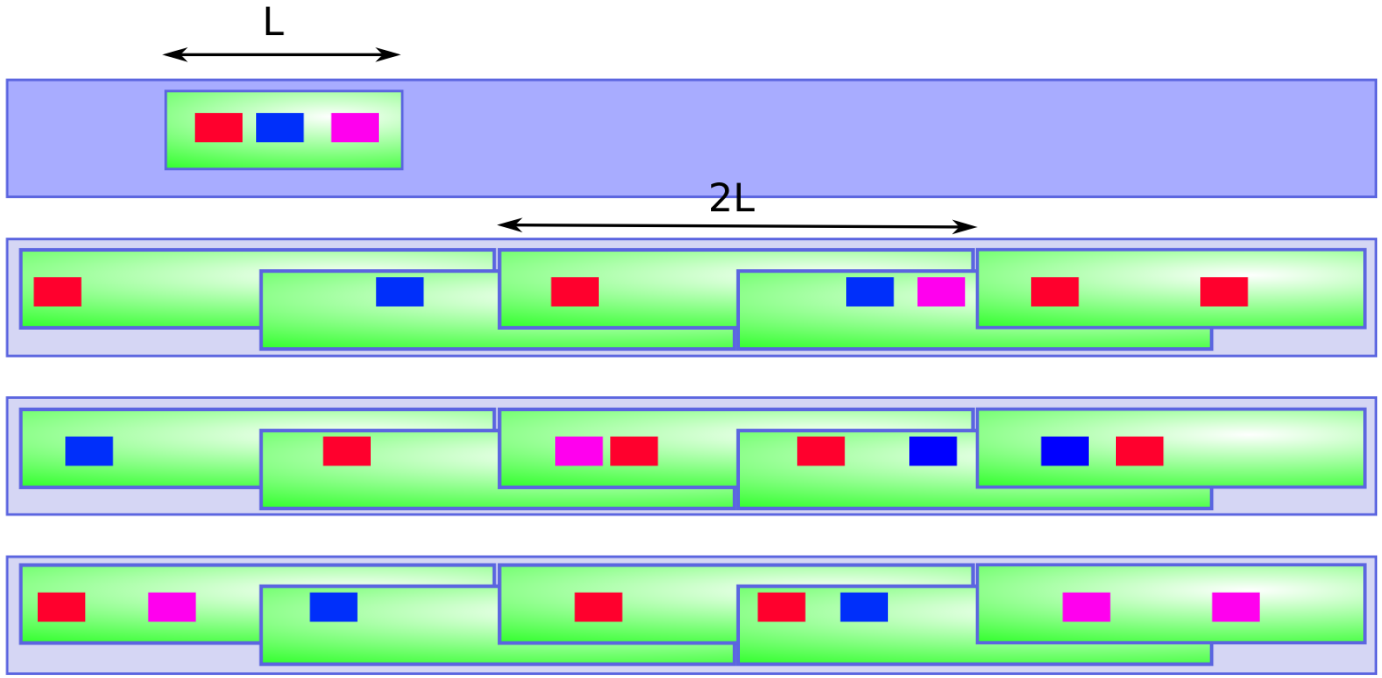
- L'**ordre** entre les k-facteurs est **conservé**



Plan de l'exposé

Algorithme - Ed'Nimbus

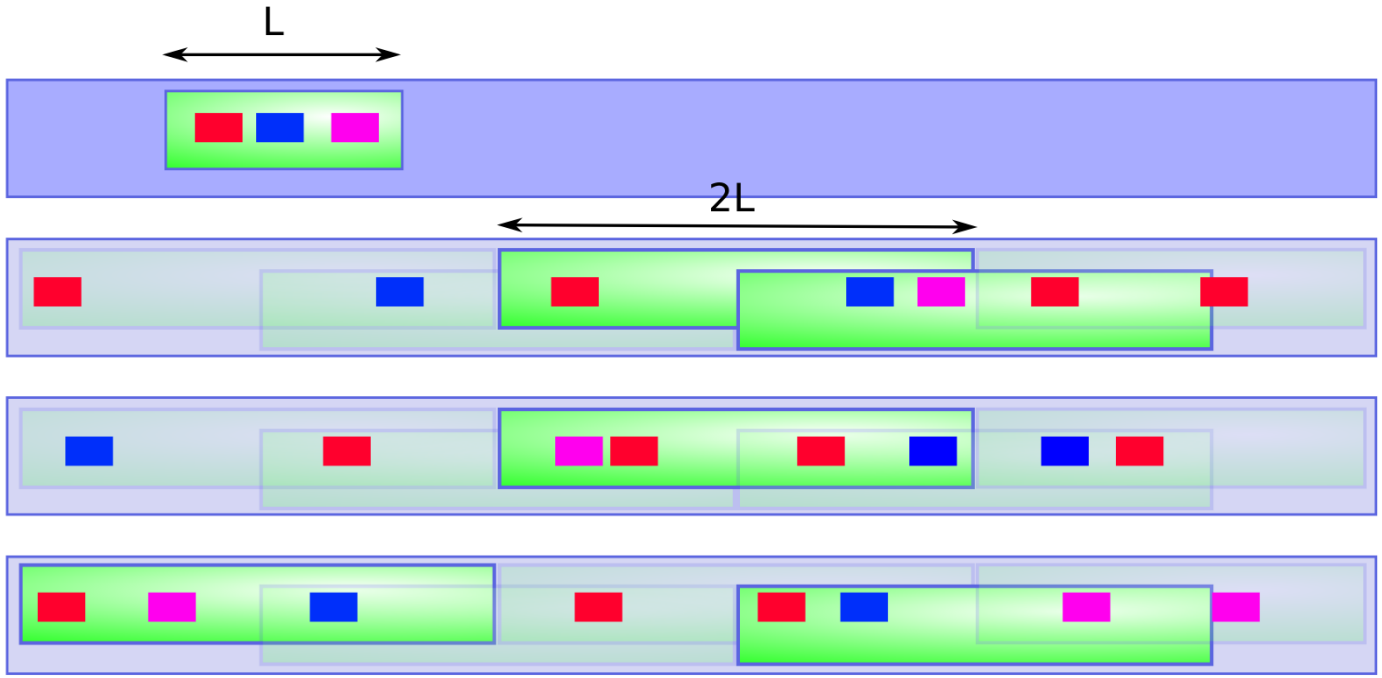
- Divise les séquences en blocs (taille $2L$ chevauchant sur L pos.)



- $r = 3$ (nombre minimum d'occurrences des répétitions)
- $p = 3$ (cond. nécessaire : nb. min. de k-facteurs partagés)

Algorithme - Ed'Nimbus

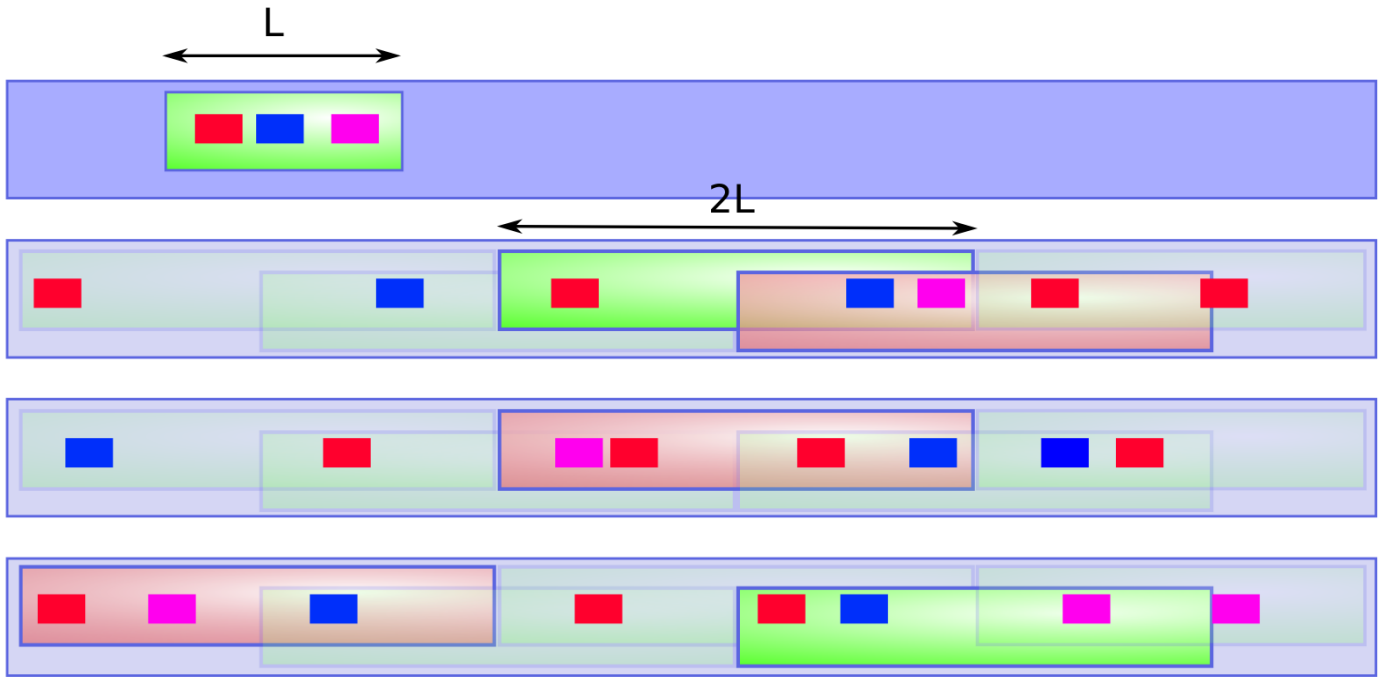
- Compte le nombre de k-facteurs partagés : *bons blocs*



- $r = 3$ (nombre minimum d'occurrences des répétitions)
- $p = 3$ (cond. nécessaire : nb. min. de k-facteurs partagés)

Algorithme - Ed'Nimbus

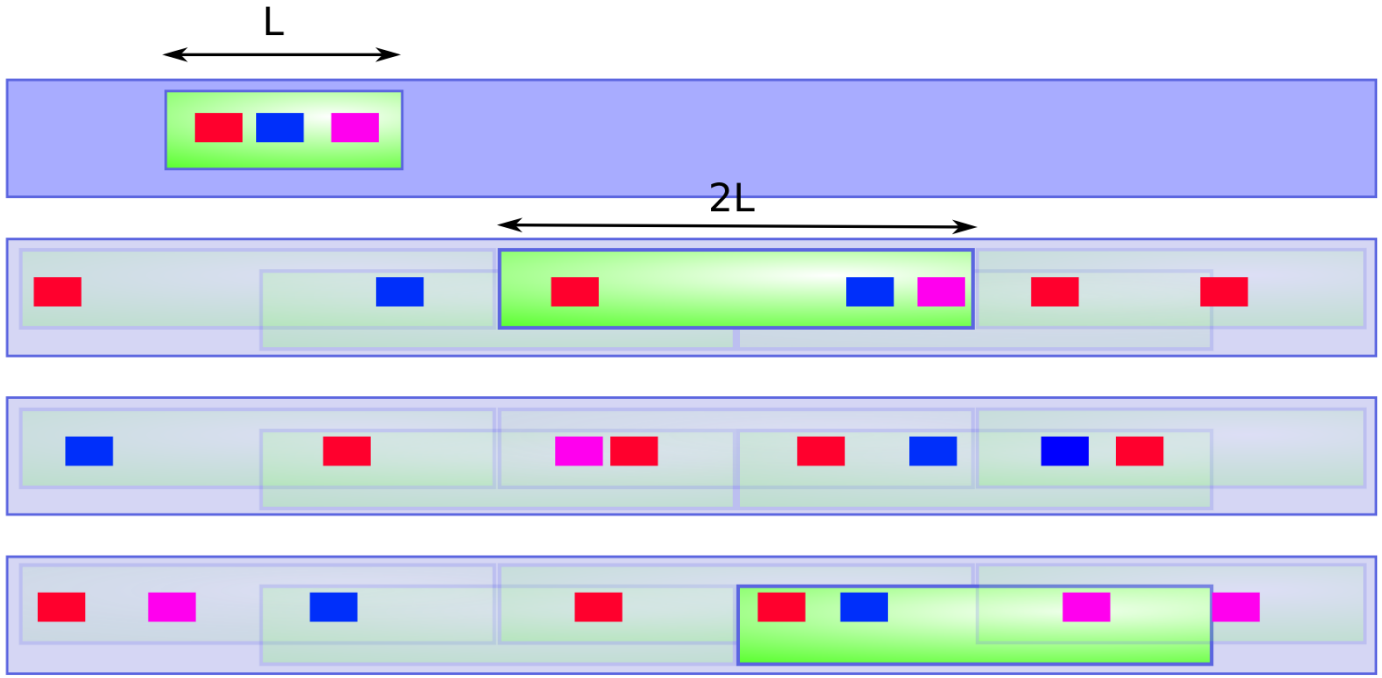
- Sur les *bons blocs* : compte le nombre de k-facteurs partagés avec ordre conservé, *blocs parfaits*



- $r = 3$ (nombre minimum d'occurrences des répétitions)
- $p = 3$ (cond. nécessaire : nb. min. de k-facteurs partagés)

Algorithme - Ed'Nimbus

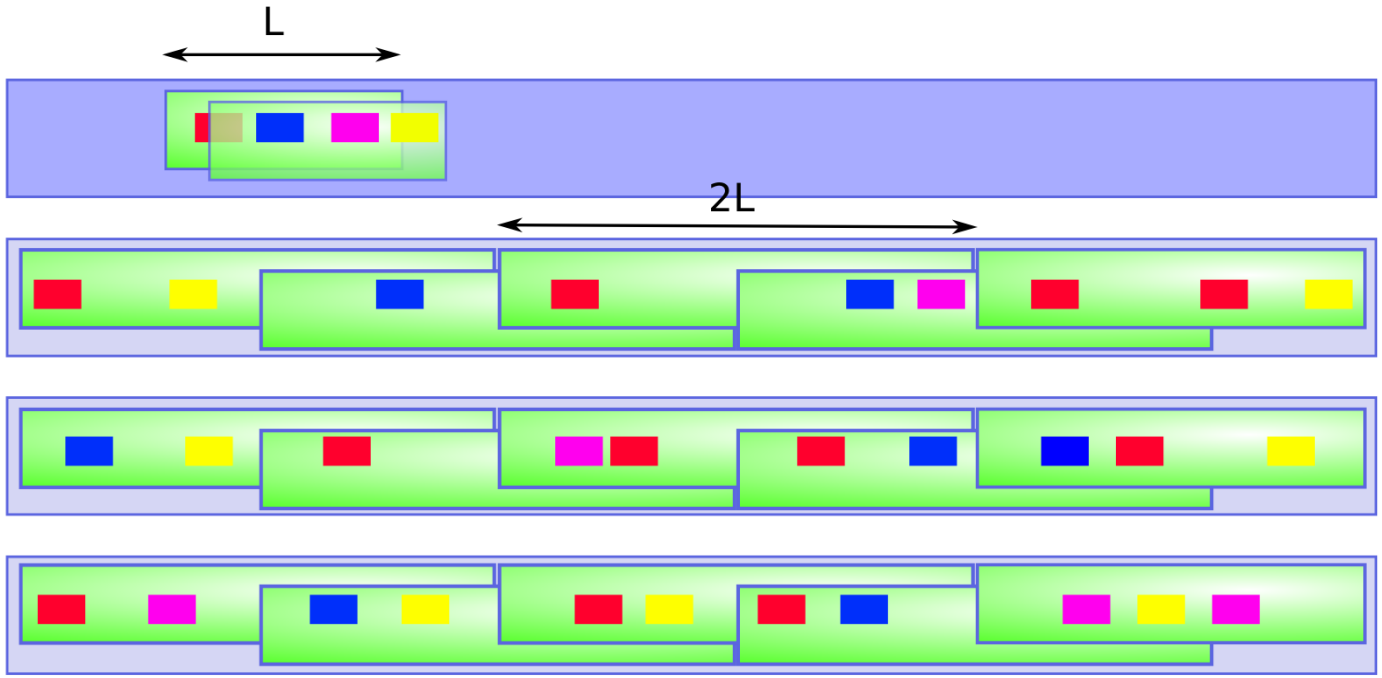
- En fonction du résultat : conserve la position de la fenêtre ?



- $r = 3$ (nombre minimum d'occurrences des répétitions)
- $p = 3$ (cond. nécessaire : nb. min. de k-facteurs partagés)

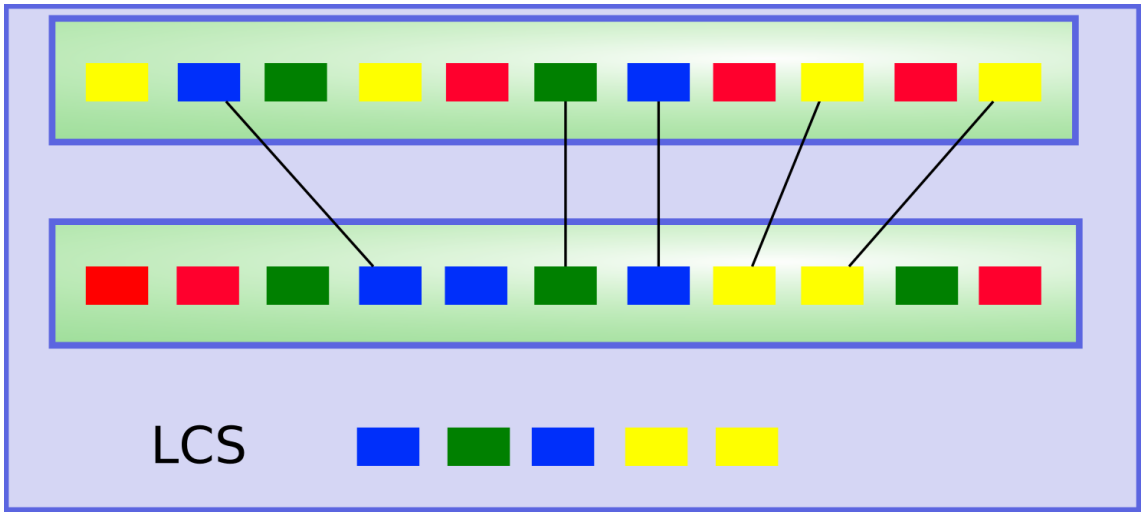
Algorithme - Ed'Nimbus

- Glisse la fenêtre d'une position

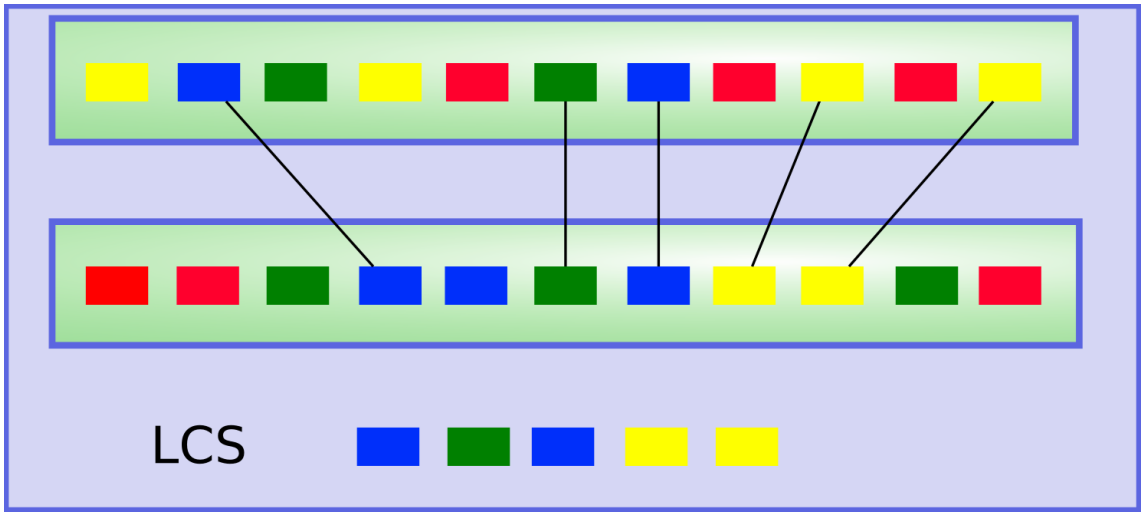


- $r = 3$ (nombre minimum d'occurrences des répétitions)
- $p = 3$ (cond. nécessaire : nb. min. de k-facteurs partagés)

Test de l'ordre des k-facteurs



Test de l'ordre des k-facteurs



J B V J R V B R J R J

R R V B B V B J J V R

LCS = B V B J J

- Gusfield

Complexité

Temps

- Cas dense

$$O\left(\frac{N^2}{L} + N \times L^2 \times \log L \times G\right)$$

- Cas épars

$$O\left(\frac{N^2}{L \times |\Sigma|^k} + N \times L \times \log L \times G\right)$$

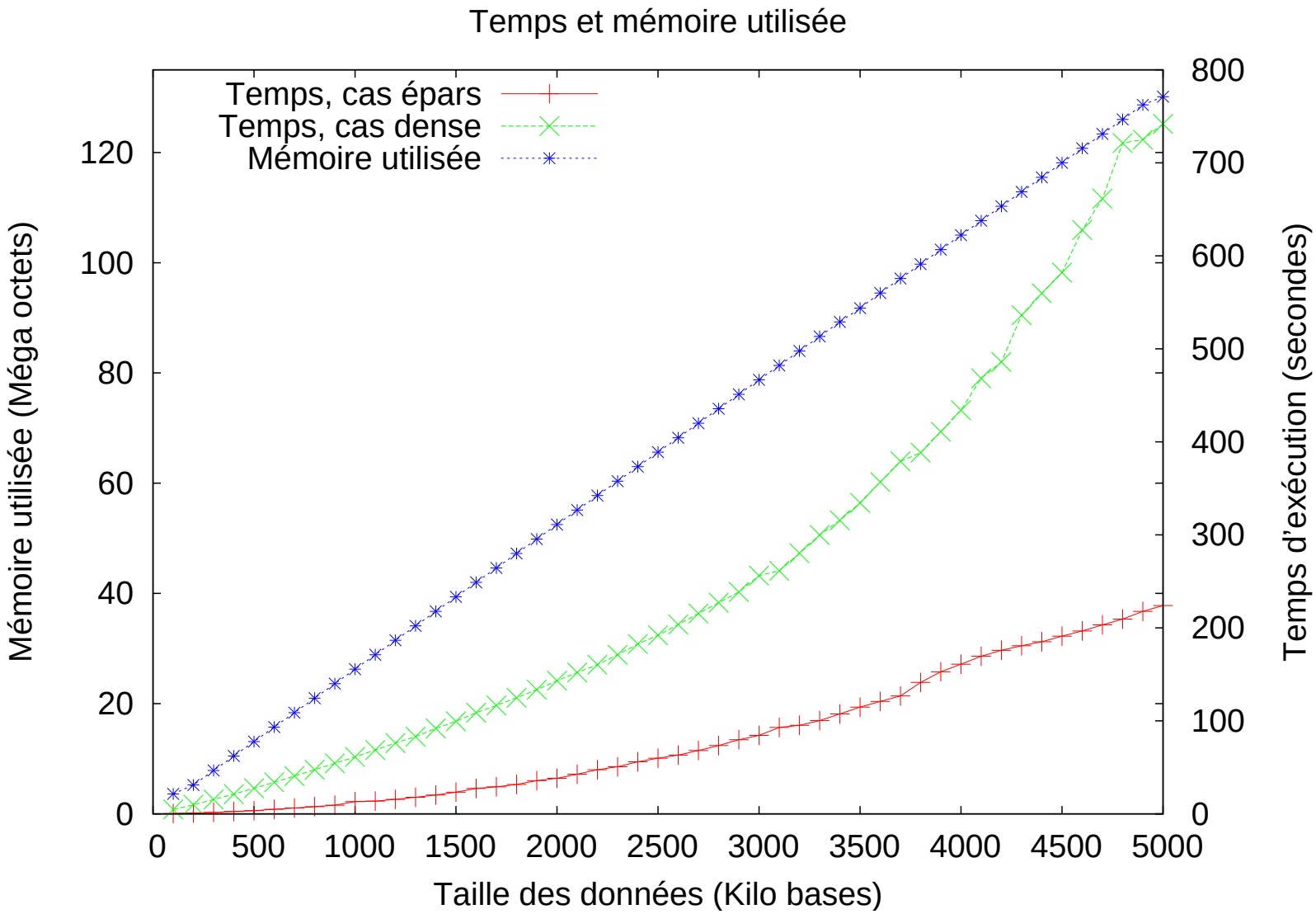
- G : Nombre de calculs de LCS
- N : Taille totale des séquences
- L : Taille des répétitions recherchées
- k : Taille des k-facteurs
- $|\Sigma|$: Taille de l'alphabet (4)

Mémoire

- $O(N)$

Plan de l'exposé

Mémoire et utilisation du temps



Estimation de la Spécificité

Taux de faux positifs

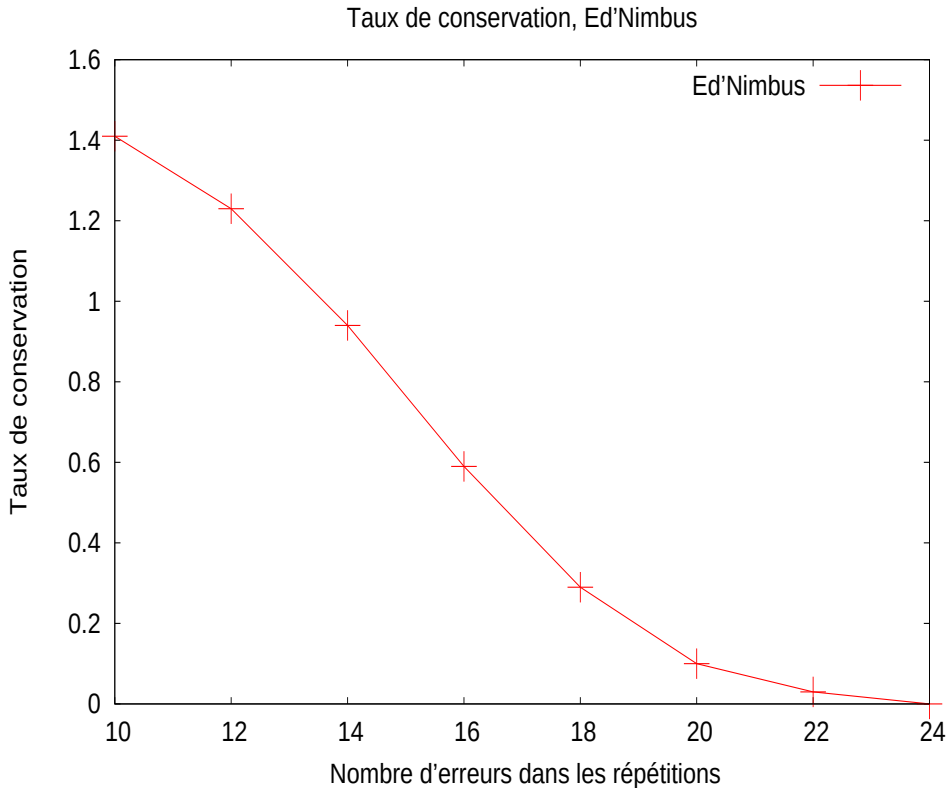
Séquence aléatoire de taille 2 Mb

- Contenant 3 répétitions (taille 100, 10 erreurs) :
  - Recherche de 3 répétitions : 0.006 %
- Contenant 100 répétitions :
  - Recherche de 100 répétitions : 0.34 %

Estimation de la spécificité

Taux de conservation

$$\frac{\text{Nombre de caractères conservés}}{\text{Nombre de caractères appartenant à une répétition}}$$



Données et paramètres

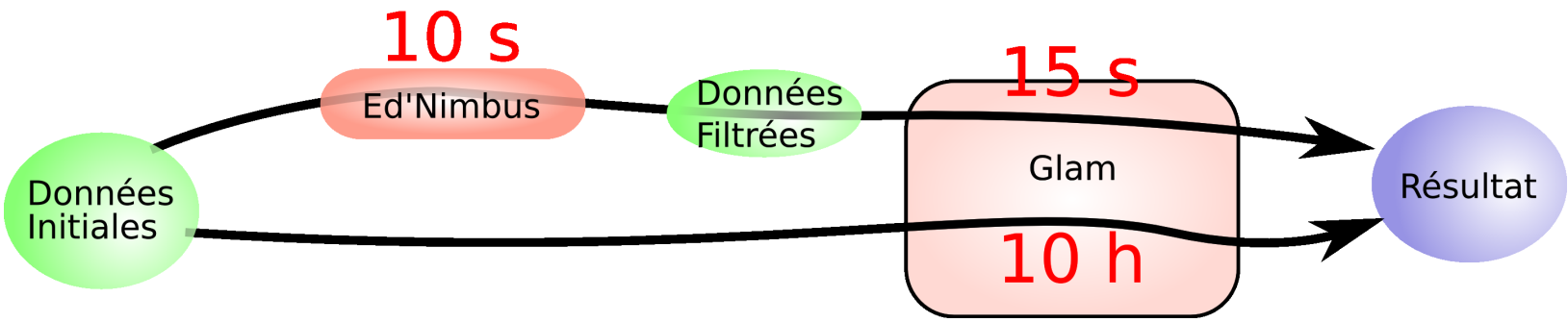
- 5 séquences aléatoires de longueur 10000
- Chaque séquence contenant une répétition de taille 100
- Paramètres : $L = 100, r = 5, d = 10, k = 5$
- Chaque test effectué 50 fois

Accélération d'algorithmes d'alignements locaux

Pré-traitement de séquences (générées) pour accélérer Glam[6]

- 5 séqs. aléatoires,
- Taille totale 1 Mb,
- Répétitions 100, 10
- Paramètres : $L = 100, d = 10, r = 5, k = 5$

- Pentium III, 1200 MHz
- 512 Mo mémoire.



- $>$ 1000 fois plus rapide

[6] Frith, M. ; Hansen, U. ; Spouge, J.L. & Weng, Z. *Finding Functionnal Sequence Elements by Multiple Local Alignement*

Exemple de requête sur le serveur d'Ed'Nimbus

`http://igm.univ-mlv.fr/~peterlon/ednimbus/`

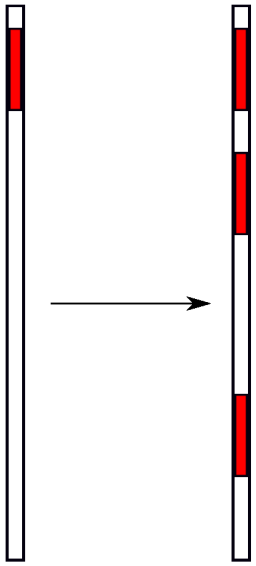
Plan de l'exposé

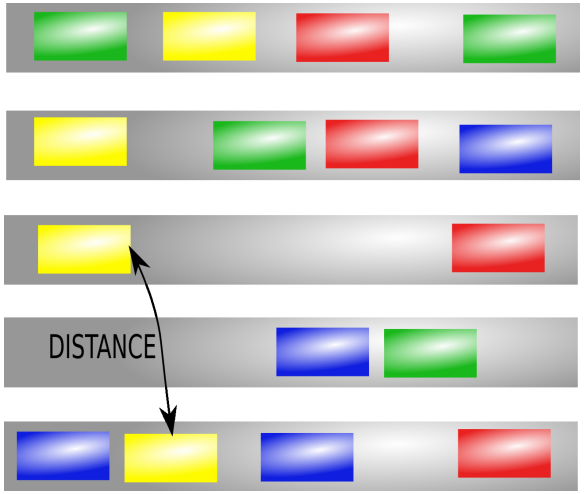
Plan de l'exposé

Conclusions

Motivations

- Apporter une solution à la détection d'éléments transposables
- Apporter une solution à la recherche de répétitions





Conclusions

Contributions

- Techniques de filtrage **exact**
  - Pour la distance de Hamming : Nimbus
  - Pour la distance d'édition : Ed'Nimbus
- Développement de structures de données
  - Arbre des bi-facteurs
  - Tableau des bi-facteurs
  - Indexation de motifs à trous
    - Arbre des suffixes
    - Automate des facteurs

Conclusions

Résultats

- Accélération d'outils
  - De recherche de répétitions approchées (Hamming ou édition)
  - D'alignement multiple local
- Bons résultats sur des séquences générées et réelles
  - Mémoire - temps
  - Spécificité
- Outil disponible et utilisable à l'adresse suivante :
  `http://igm.univ-mlv.fr/~peterlon/ednimbus/`

Plan de l'exposé

Perspectives

Tests

- Multiplier les tests sur des séquences réelles
- Analyse précise des résultats biologiques

Perspectives

Amélioration des résultats

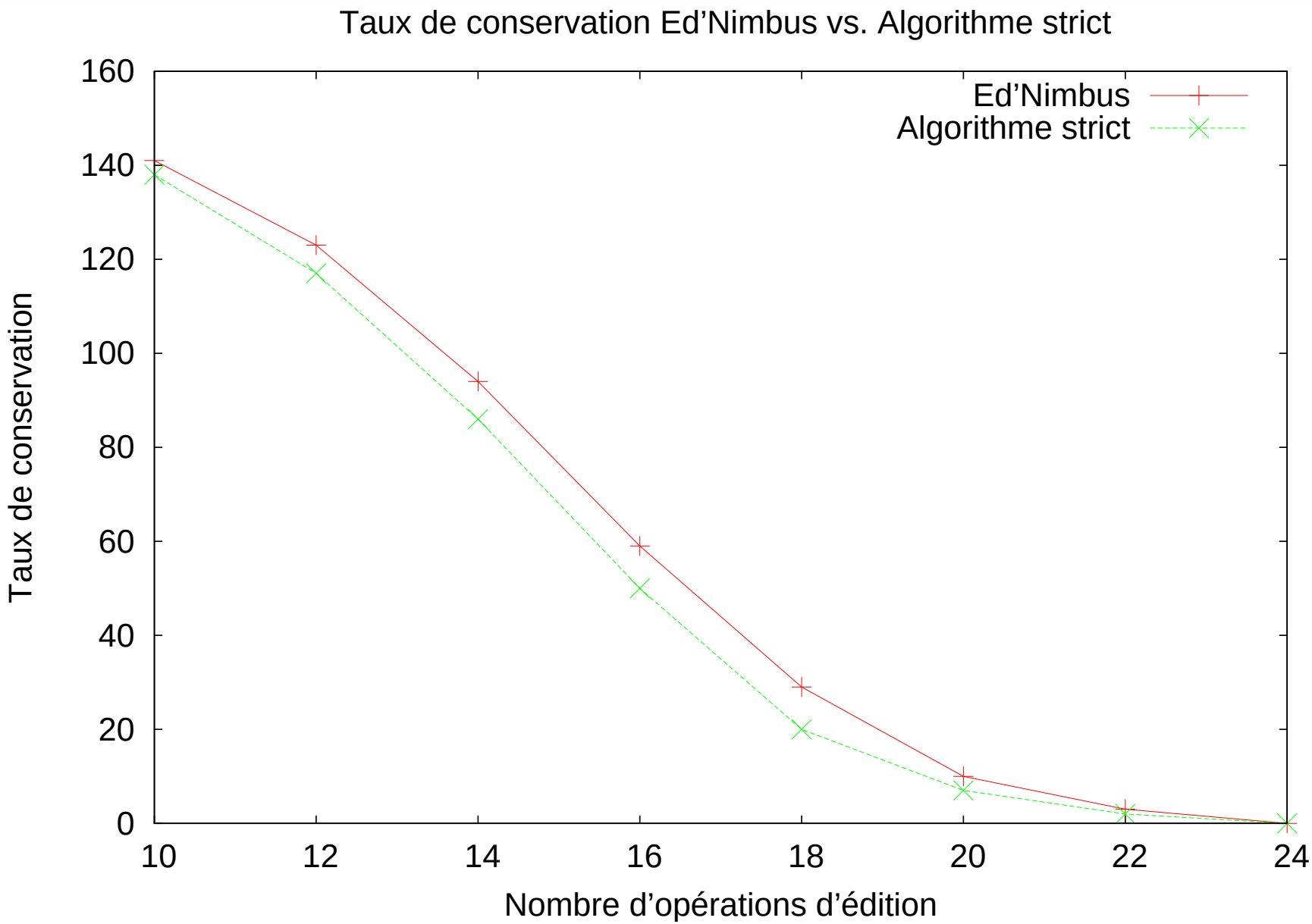
- Estimation de la meilleure valeur de k en fonction des paramètres,
  - *k petit* :
    - p (condition sur le nombre de k-facteurs partagés) grands
    - Probabilité de trouver les k-facteurs partagés plus grande
  - *k grand* :
    - p (condition sur le nombre de k-facteurs partagés) petit
    - Probabilité de trouver les k-facteurs partagés plus petite

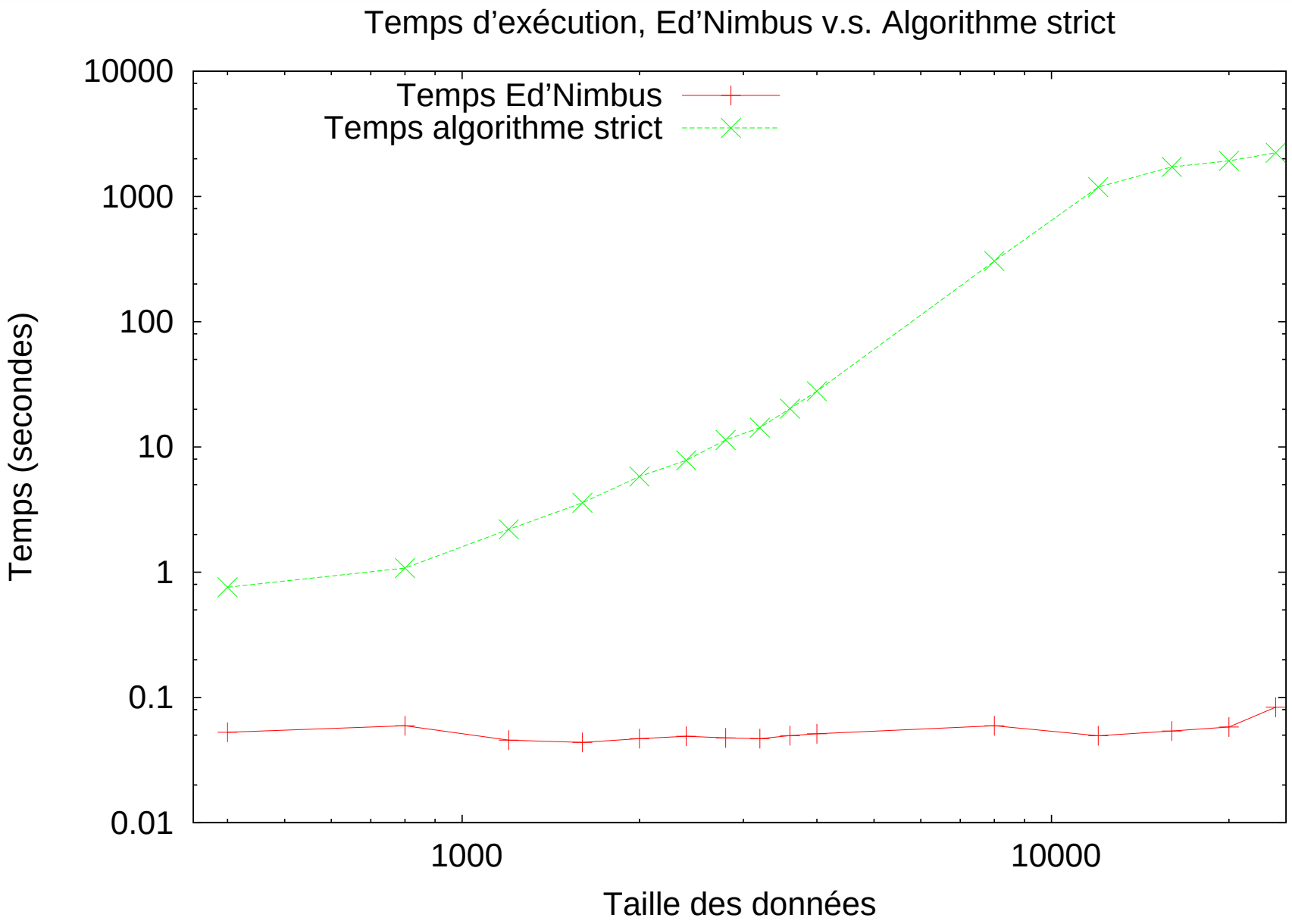
Perspectives

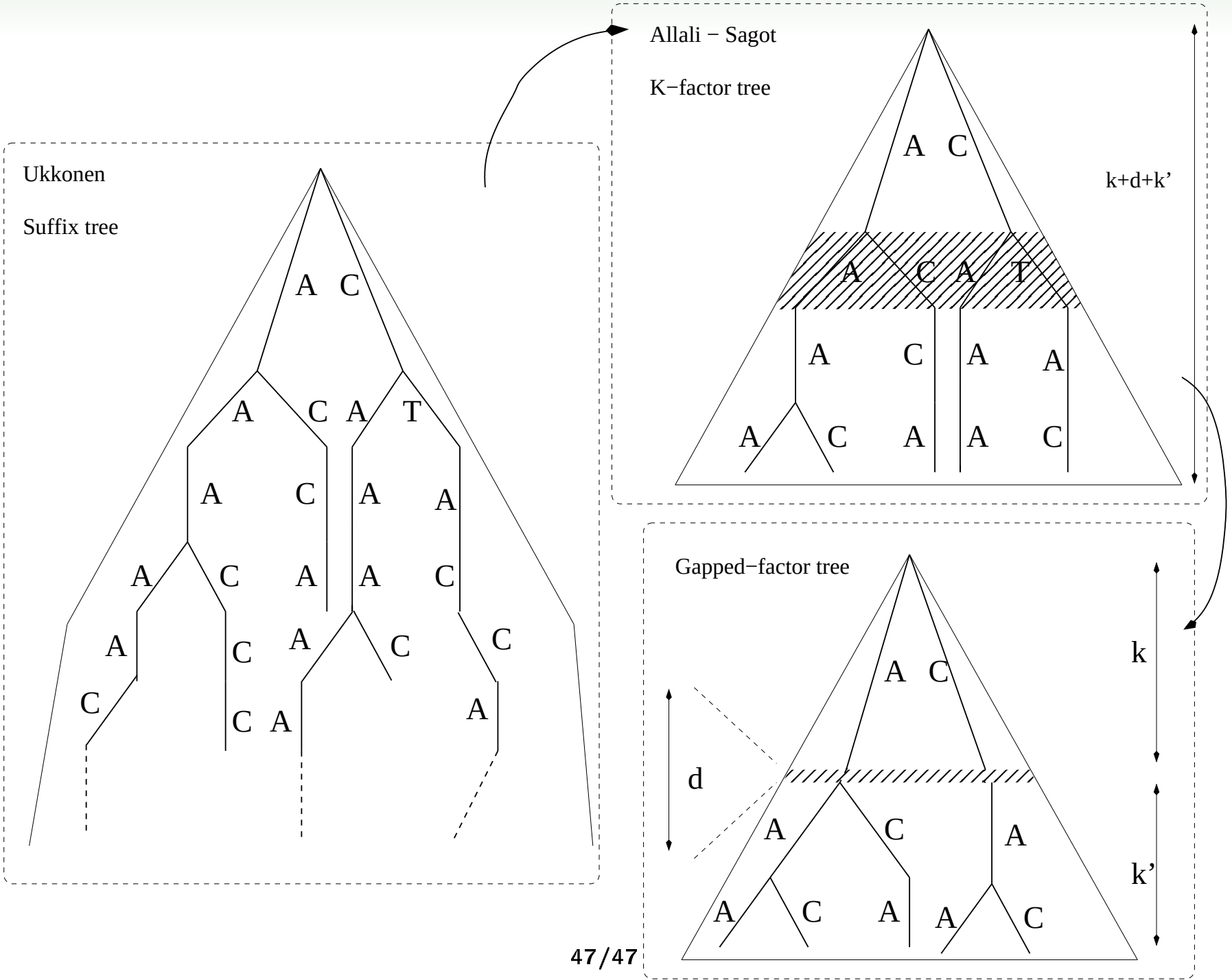
Modification des techniques

- Développer un logiciel complet de détection de répétitions
  - Filtrage
  - Alignement multiple adapté aux données d'Ed'Nimbus
  - Visualisation
- Filtrage pour des répétitions de taille $L \geq L_0$
- Utilisation de k-facteurs approchés (autorisant une ou deux substitutions)

merci


Taux de conservation Ed'Nimbus vs. Algorithme strict
Ed'Nimbus
Algorithme strict
Taux de conservation
Nombre d'opérations d'édition
0
20
40
60
80
100
120
140
160
10
12
14
16
18
20
22
24


Temps d'exécution, Ed'Nimbus v.s. Algorithme strict
Temps Ed'Nimbus
Temps algorithme strict
Temps (secondes)
10000
1000
100
10
1
0.1
0.01
1000
10000
Taille des données


Ukkonen
Suffix tree
Allali − Sagot
K−factor tree
k+d+k'
Gapped−factor tree
d
k
k'

LCP	Pos	Suffix
0	5	AACC...
2	52	AAGC...
2	1	AATA...
3	23	AATG...
3	31	AATT...
4	36	AATT...
0	18	CAAT...
4	29	CAAT...
0	5	GAAA...
2	8	GACA...
1	91	GGAG...
3	86	GGAT...
0	12	TCAT...
2	7	TCGA...
3	94	TCGC...
2	17	TCTT...
4	4	TCTT...

Name	Pos
0	5
0	52
0	1
0	23
0	31
0	36
1	18
1	29
2	5
2	8
3	91
3	86
4	12
4	7
4	94
4	17
4	4

Pred Name	Name	Pos
2	0	5
5	0	52
-	0	1
2	0	23
8	0	31
6	0	36
4	1	18
0	1	29
8	2	5
14	2	8
5	3	91
3	3	86
3	4	12
2	4	7
5	4	94
5	4	17
9	4	4

Name	Pos
0,0	4 -> 12
0,1	29 -> 13
0,2	6 -> 123
0,11	91 -> 24
1,4	21
1,8	1 -> 32
1,19	8 -> 18
2,0	5 -> 23
2,4	7 -> 11
2,8	74 -> 81
2,12	2->31->0
3,0	52 -> 2
3,1	79 -> 4
3,8	2-> 7 -> 9
3,9	167 -> 86
4,0	71 -> 43
4,1	28 -> 1