

# Filtrage de séquences d'ADN pour la recherche de longues répétitions multiples Ed'Nimbus, et quelques structures de données

Pierre Peterlongo

Irisa

## Plan de l'exposé

Présentation (**biologique**) du problème

Présentation (**algorithmique**) du problème

Nimbus

Ed'Nimbus

Algorithme

Quelques résultats

Structures de données utilisées

Arbre des bi-facteurs

Tableau des bi-facteurs

Conclusions

# Plan de l'exposé

Présentation (**biologique**) du problème

Présentation (**algorithmique**) du problème

Nimbus

Ed'Nimbus

Algorithme

Quelques résultats

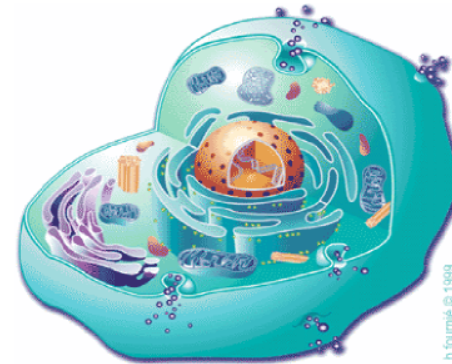
Structures de données utilisées

Arbre des bi-facteurs

Tableau des bi-facteurs

Conclusions

# ADN et Synthèse protéique

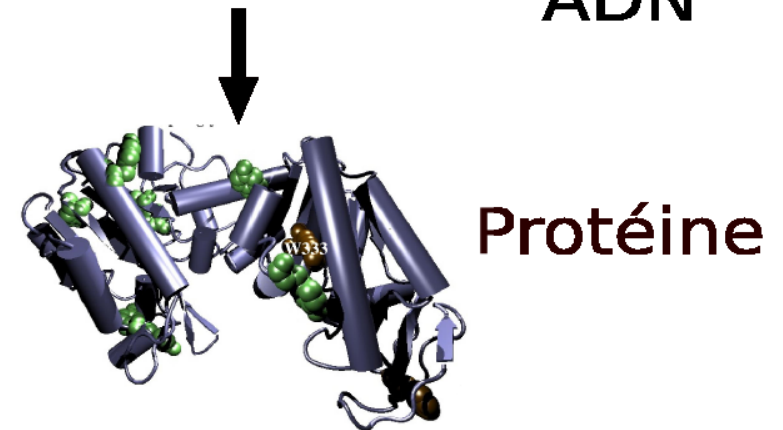
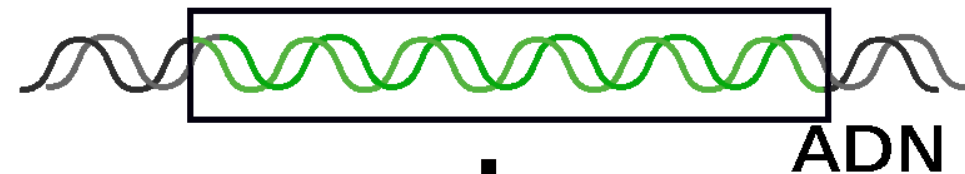


- Gène (Segment d'ADN codant pour une protéine)
- ARN
- Protéine (Constituant principal des cellules)

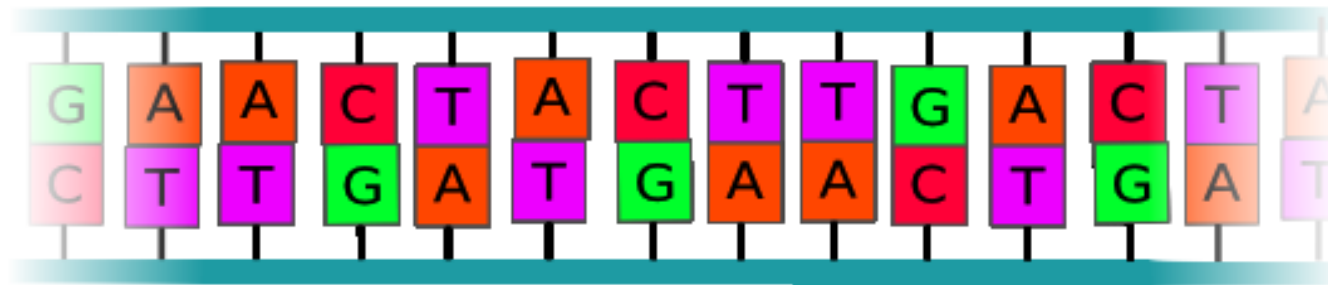
# ADN et Synthèse protéique



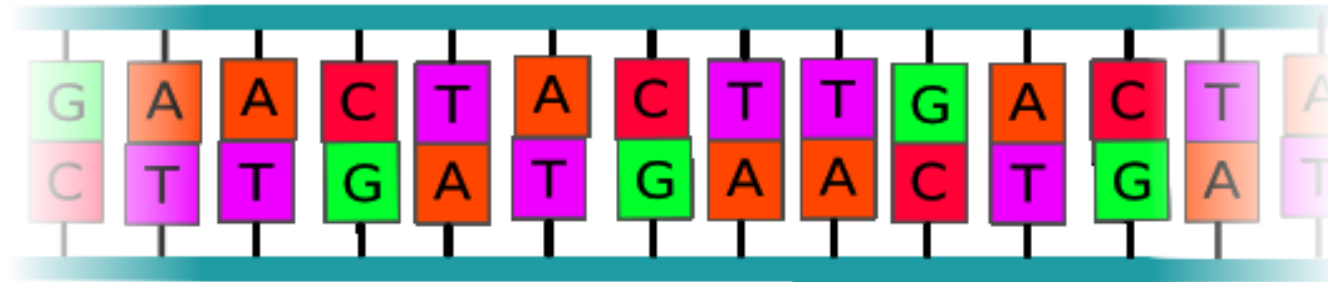
- Gène (Segment d'ADN codant pour une protéine)
- ARN
- Protéine (Constituant principal des cellules)



# Composition de l'ADN

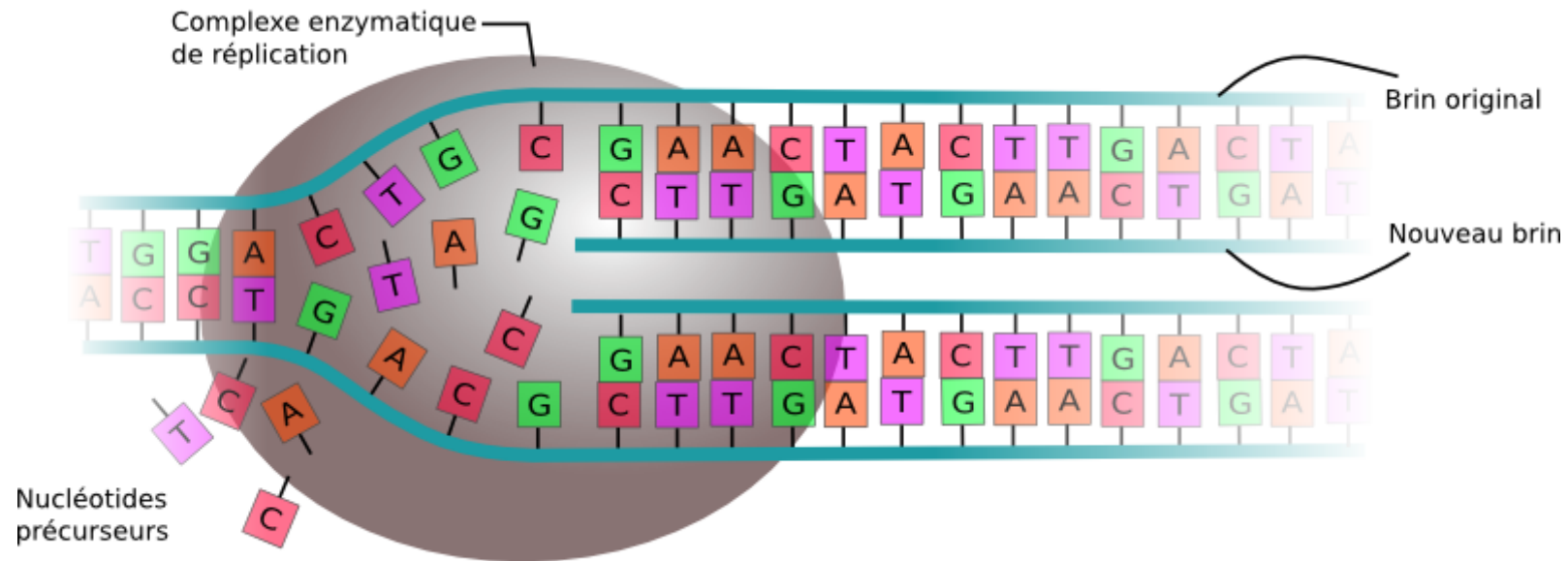


# Composition de l'ADN



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

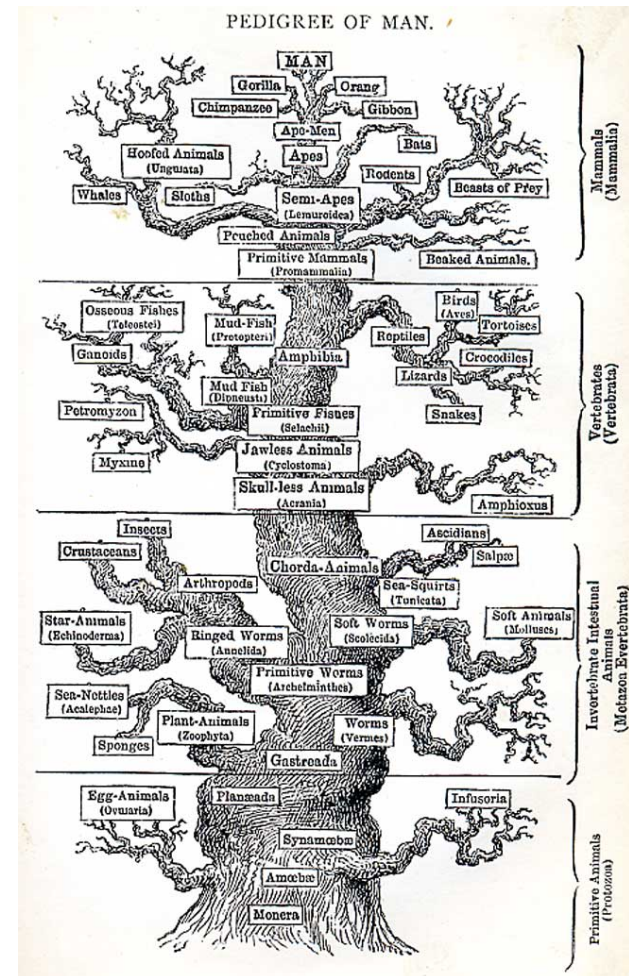
- Aide à l'annotation des génomes (transfert de connaissance)



# Similarités - Répétitions

## Utilisation

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



## Similarités - Répétitions

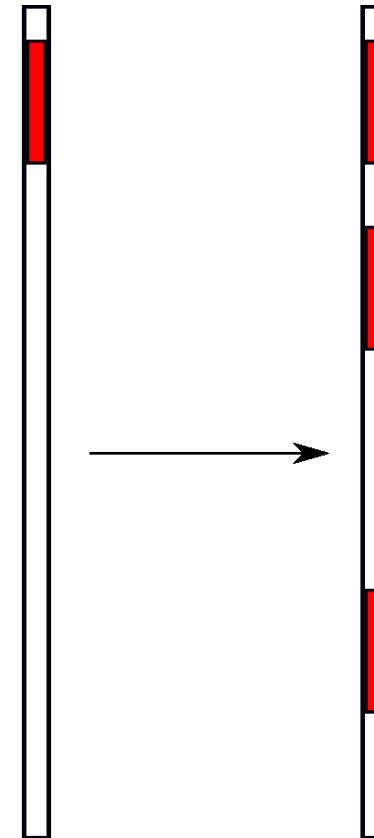
### Répétitions sur les génomes - Éléments transposables

- 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  $\geq 2$
- Dans de (très) grandes séquences :  $> 1$  Mb

## Plan de l'exposé

Présentation (**biologique**) du problème

Présentation (**algorithmique**) du problème

Nimbus

Ed'Nimbus

Algorithme

Quelques résultats

Structures de données utilisées

Arbre des bi-facteurs

Tableau des bi-facteurs

Conclusions

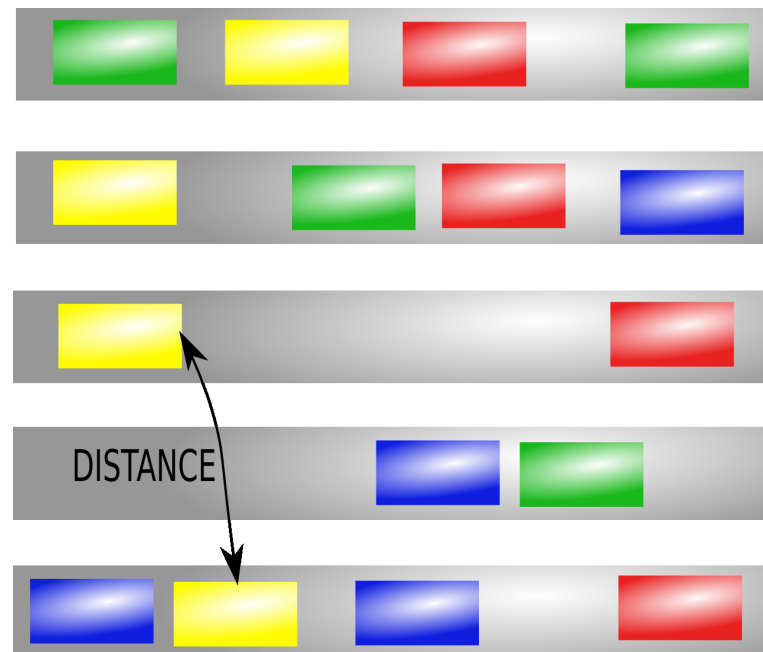
## Motivations

- 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

### Solution existante

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

ACTCGCA
ACTTGCA
A-TCGCA
ACTTG-A

# Motivations

## Limites de l'alignement multiple

### Complexité trop importante

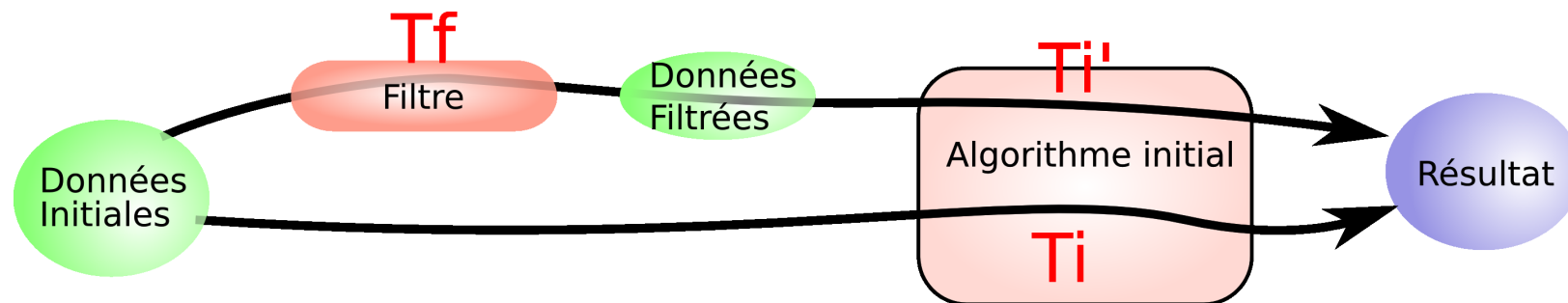
- Temps d'exécution  
 $O(2^{\text{nombre de séquences}} \times \text{taille des séquences}^{\text{nombre de séquences}})$
- Trop long, par exemple :
  - Ordinateur *classique*,  $10^9$  opérations par secondes
  - 3 séquences de taille 1 mégabase : **plusieurs années** de calcul
- Il existe des solutions,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



## Solution proposée

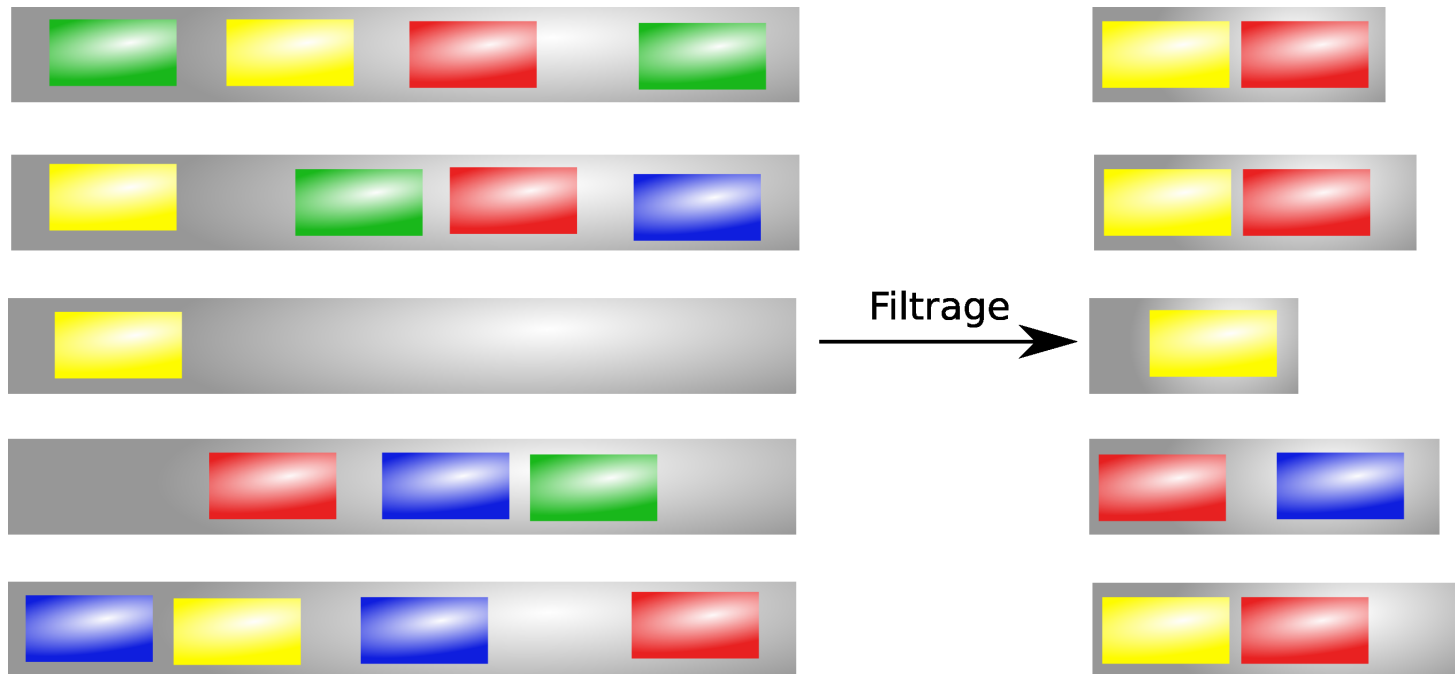
### Filtre

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

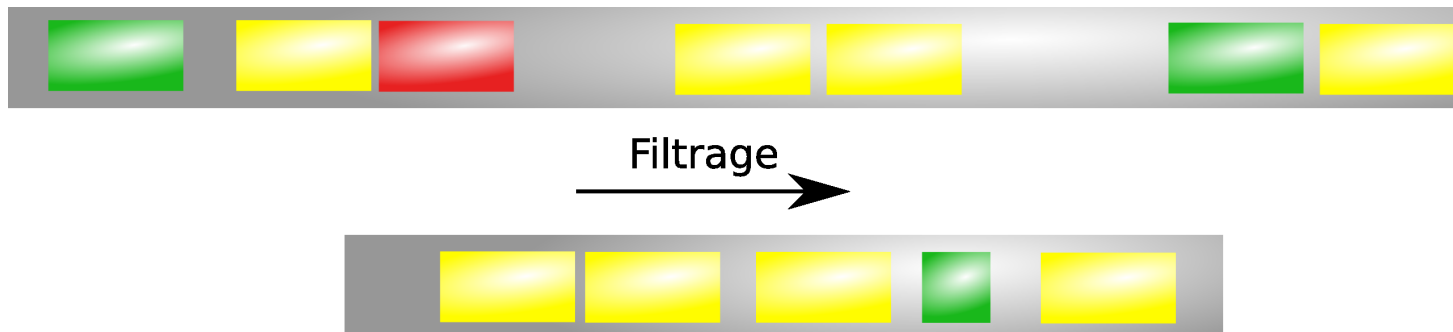


- $T_f + T_{i'} \ll T_i$

- 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 <sup>1</sup> , Swift <sup>2</sup>	∅
Non exact	Yass <sup>3</sup>	Multi-lagan <sup>4</sup>

---

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

<sup>2</sup>Rasmussen, K.R. ; Stoye, J. ; Myers, E.W. *Efficient q-Gram Filters for Finding All  $\epsilon$ -Matches Over a Given Length*

<sup>3</sup>Noé, L. ; Kucherov, G. *YASS : enhancing the sensitivity of DNA similarity search*

<sup>4</sup>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.

ATTAAATAATT  
ATAAATAATT

## 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*

ATTAAATAATTT  
ATAAATAATTT

## Plan de l'exposé

Présentation (**biologique**) du problème

Présentation (**algorithmique**) du problème

**Nimbus**

Ed'Nimbus

Algorithme

Quelques résultats

Structures de données utilisées

Arbre des bi-facteurs

Tableau des bi-facteurs

Conclusions

## 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

A T T A A A A T T T T  
 A T A A A T A T T T T

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

A T T A A A A T T T T  
 A T A A A T A T T T T  
 A T A A A T A T A T

### Condition nécessaire - $r$ mots

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

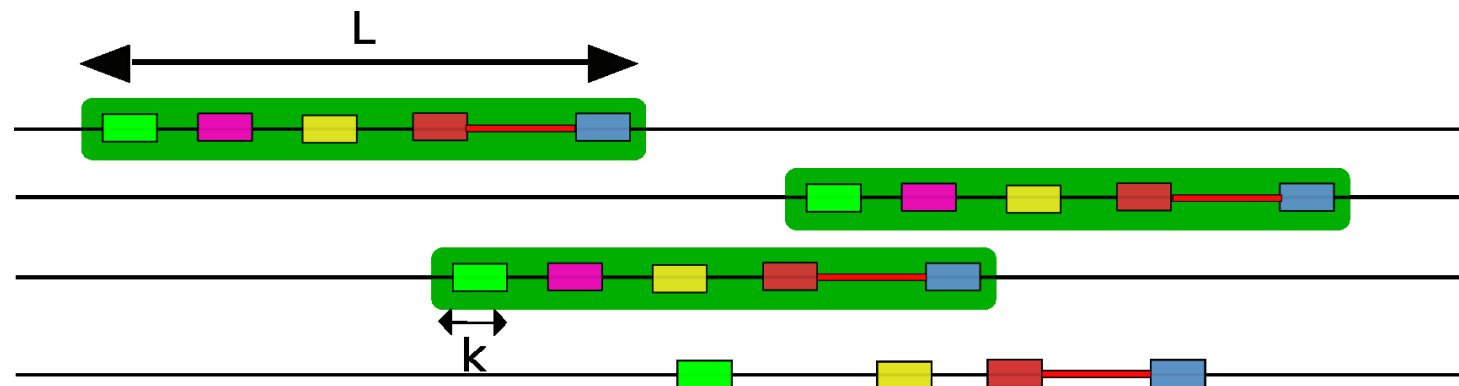
$$p = \left\lfloor \frac{L}{k} \right\rfloor - d - (r - 2) \left\lfloor \frac{d}{2} \right\rfloor$$

$k$ -facteurs non chevauchant

# Application - Idée générale

## Détection d'ensembles de zones

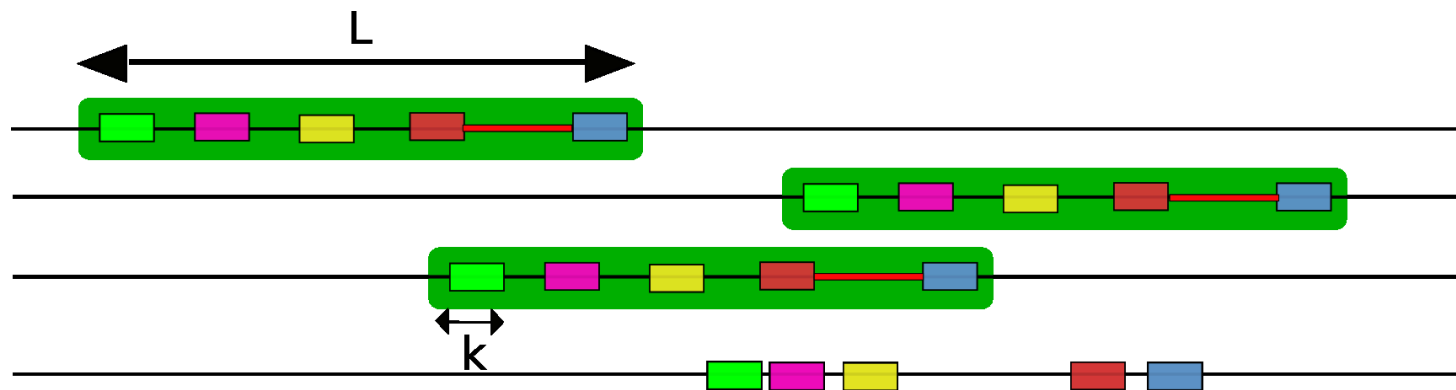
- De taille  $L$
- Apparaissant au moins dans  $r$  séquences
- Partageant au moins  $p(= 5)$   $k$ -facteurs



# 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**



## Limites de Nimbus

- Complexité importante
- 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

Présentation (**biologique**) du problème

Présentation (**algorithmique**) du problème

Nimbus

**Ed'Nimbus**

Algorithme

Quelques résultats

Structures de données utilisées

Arbre des bi-facteurs

Tableau des bi-facteurs

Conclusions

## 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

ATTAAATT  
ATAAATT



Nombre de  $k$ -facteurs (possiblement chevauchants )  
partagés

ATTAAATT  
ATAAATT

Condition nécessaire

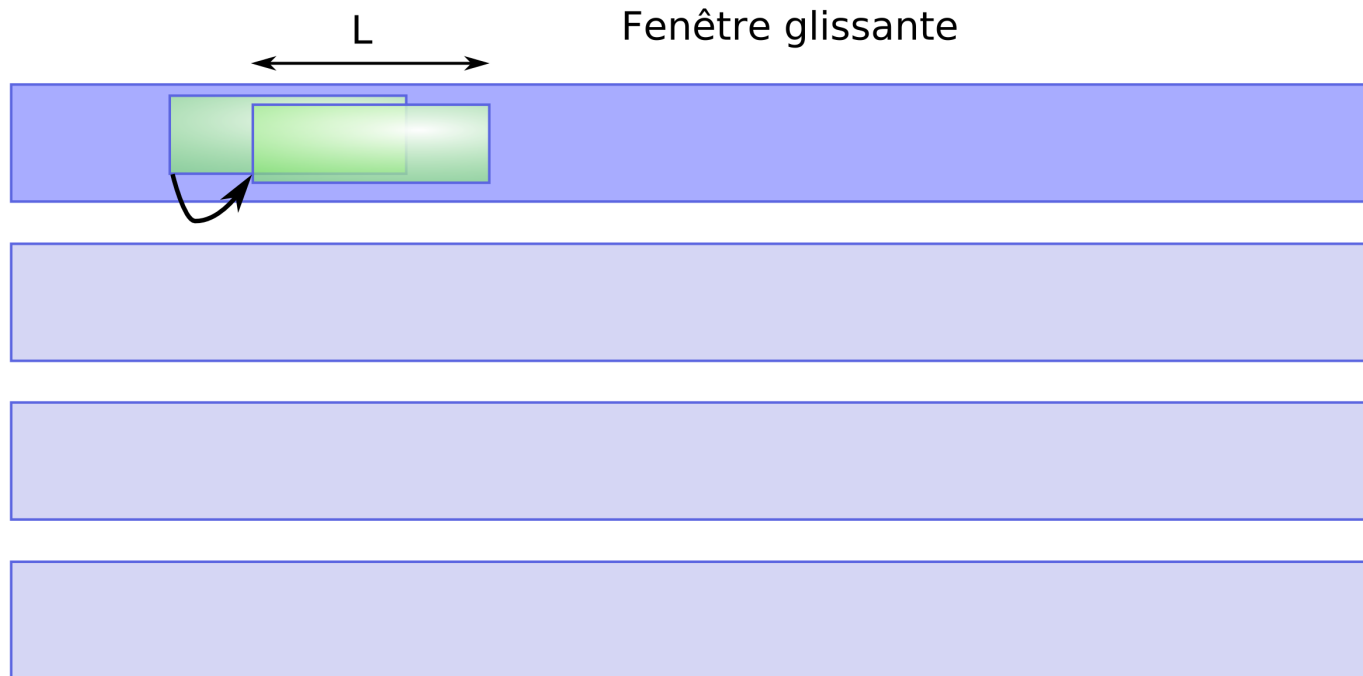
- Deux mots de taille  $L$
- Distant d'au plus  $d$  opérations d'édition
- Partagent au moins

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

$k$ -facteurs

# Application - Idée générale

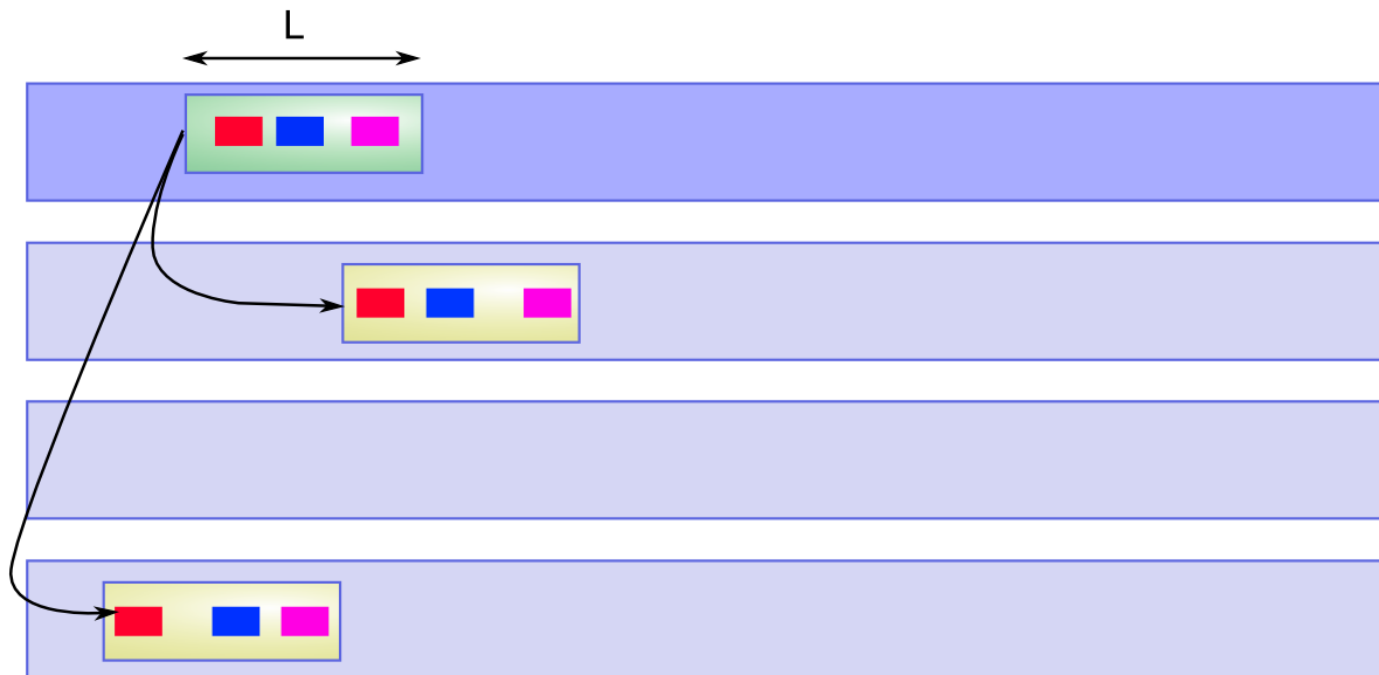
Sur chaque séquence, une fenêtre **glisse** pour tester si elle peut être une répétition



## Application - Idée générale

Toute paire respecte la condition sur  $p$

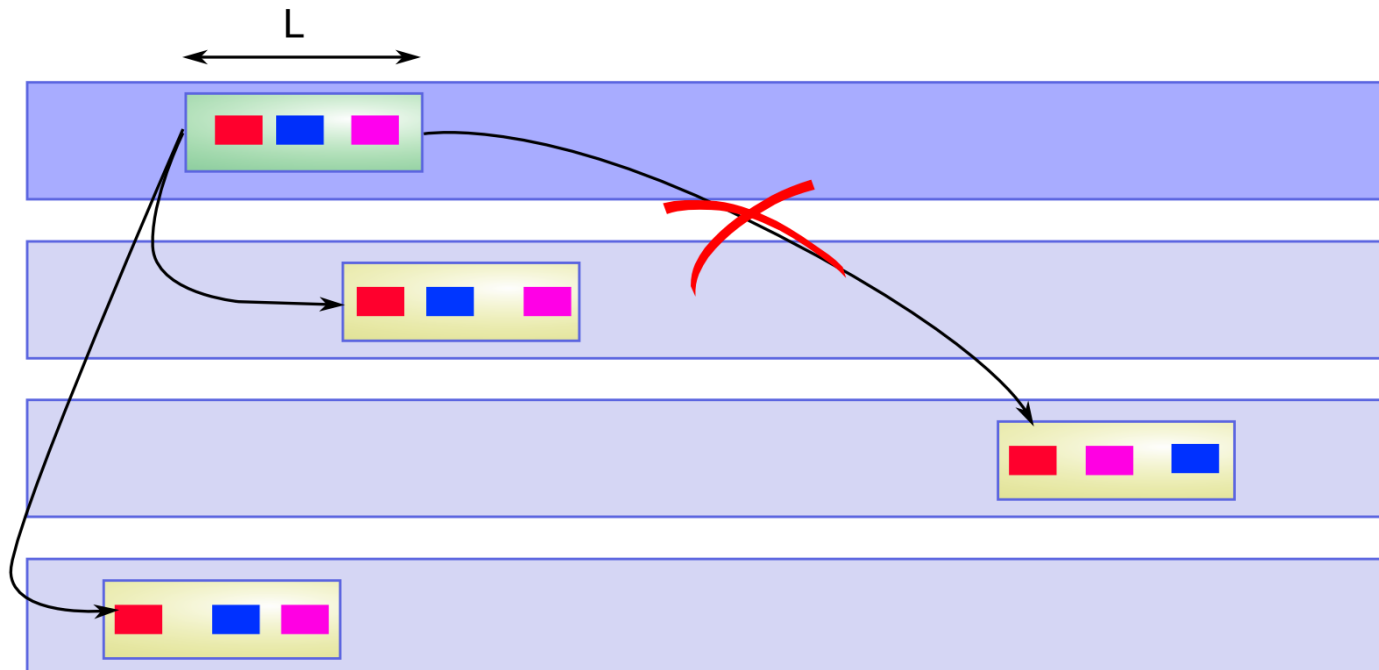
- $p = 3$  (nombre minimum de  $k$ -facteurs partagés)



## Application - Idée générale

Toute paire respecte la condition sur  $p$

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



## Plan de l'exposé

Présentation (**biologique**) du problème

Présentation (**algorithmique**) du problème

Nimbus

**Ed'Nimbus**

Algorithme

Quelques résultats

Structures de données utilisées

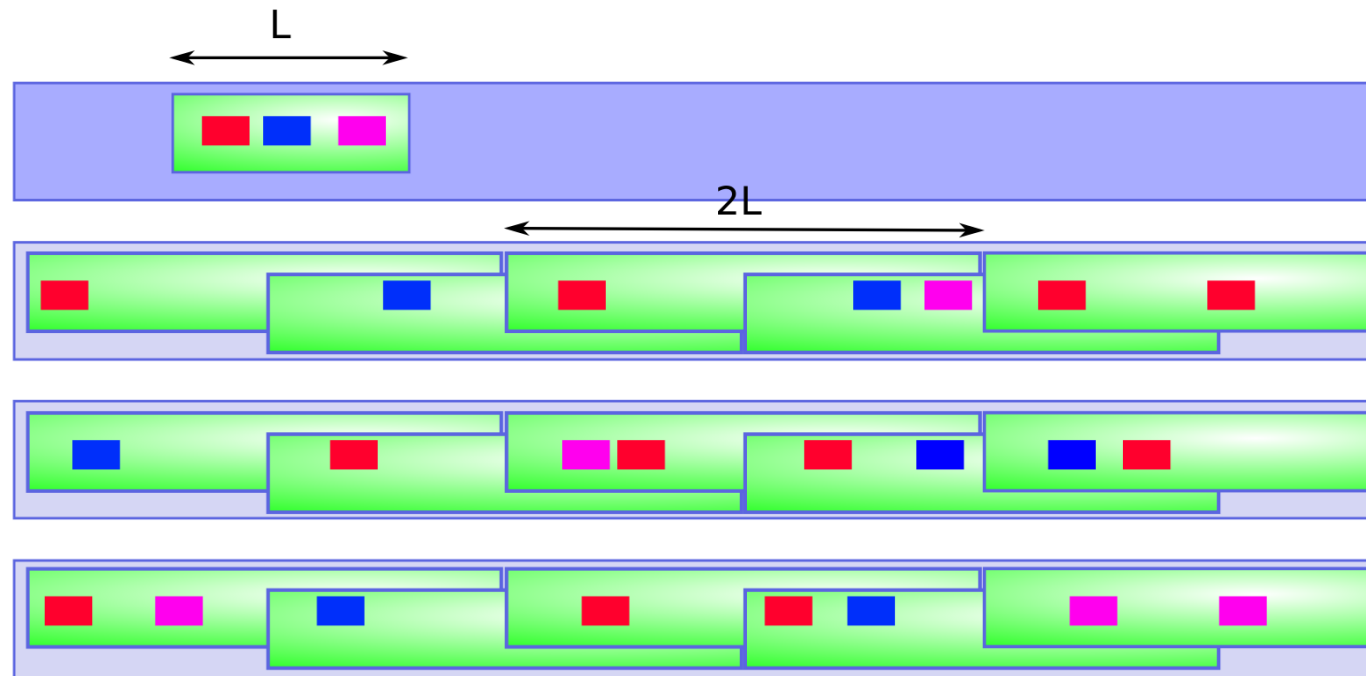
Arbre des bi-facteurs

Tableau des bi-facteurs

Conclusions

## Algorithme - Ed'Nimbus

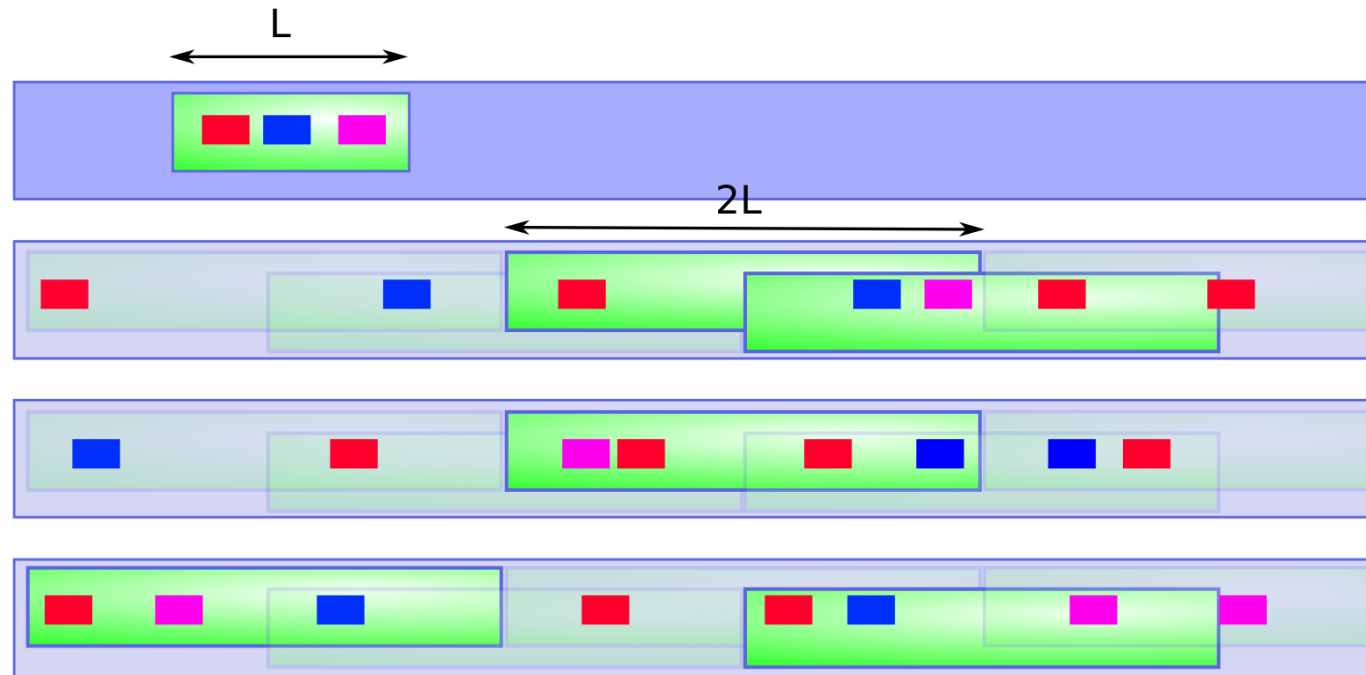
- Divise les séquences en blocs (taille  $2L$  chevauchant sur  $L$  pos.)



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

## Algorithme - Ed'Nimbus

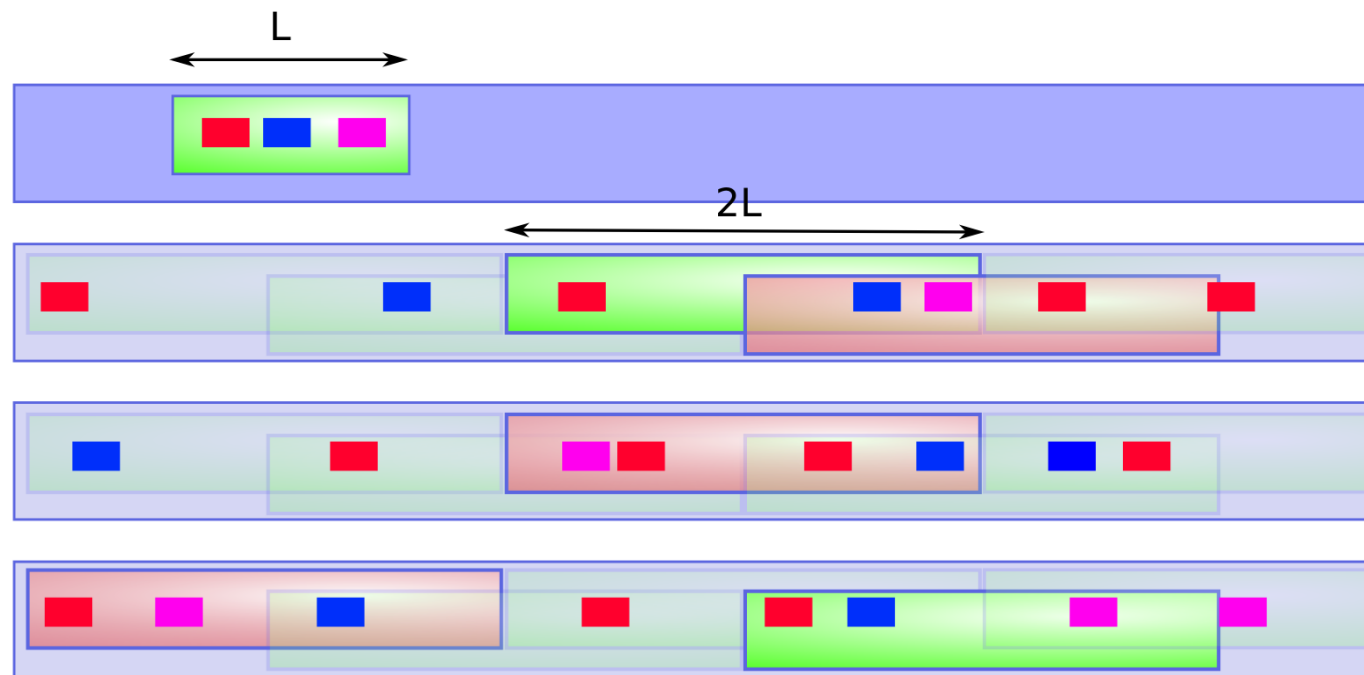
- Compte le nombre de  $k$ -facteurs partagés : *bons blocs*



- $r = 3$  (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*

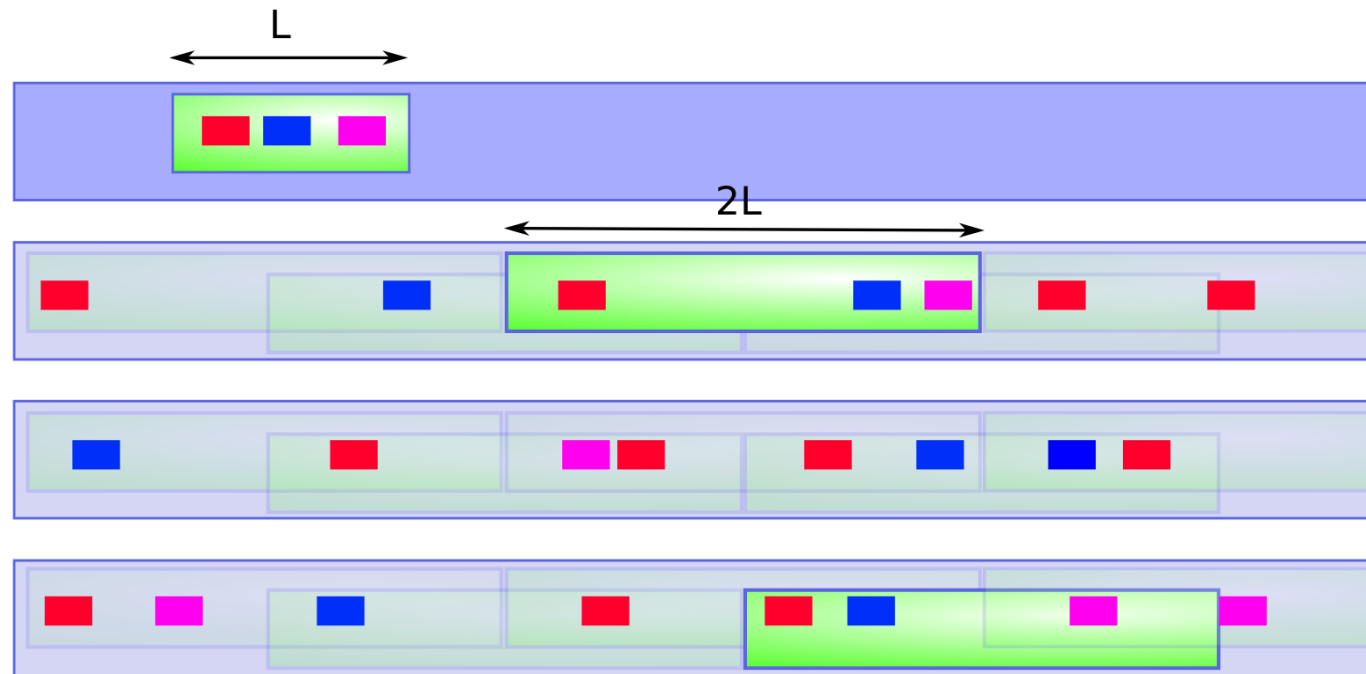


- $r = 3$  (nombre minimum d'occurrences des répétitions)
- $p = 3$  (cond. nécessaire : nb. min. de  $k$ -facteurs partagés)



# Algorithme - Ed'Nimbus

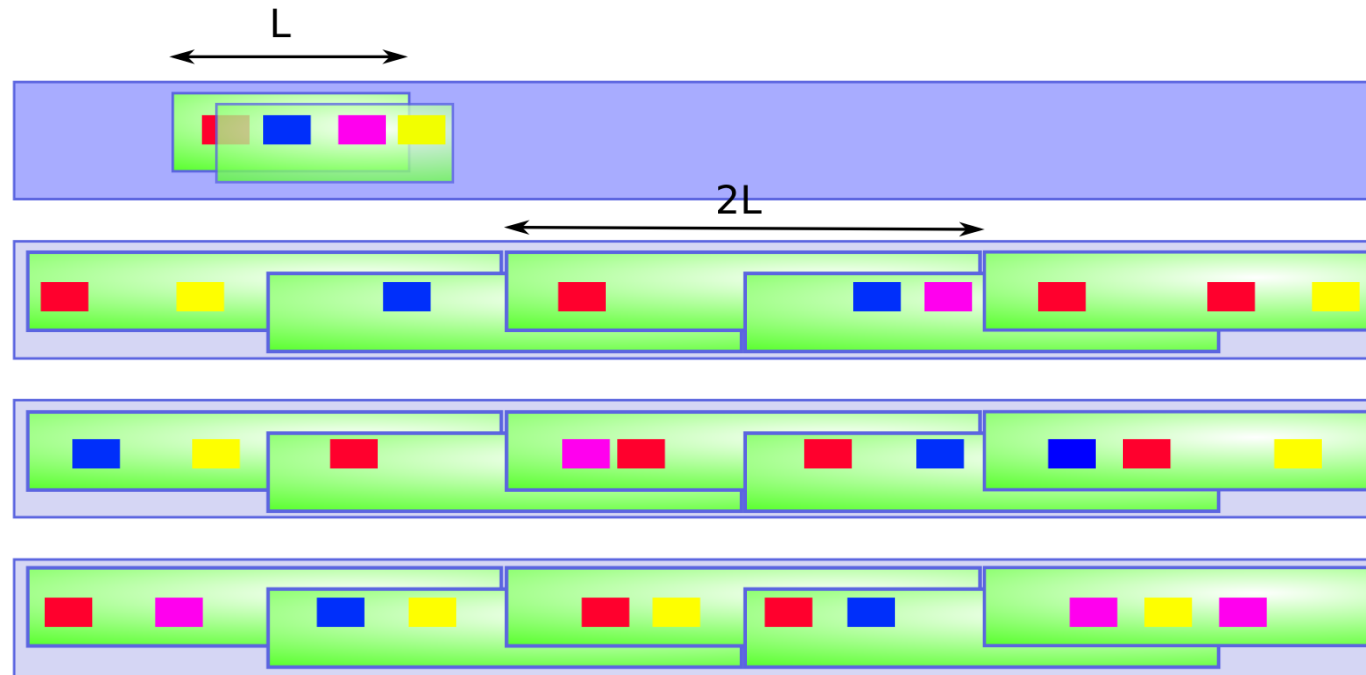
- En fonction du résultat : conserve la position de la fenêtre ?



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

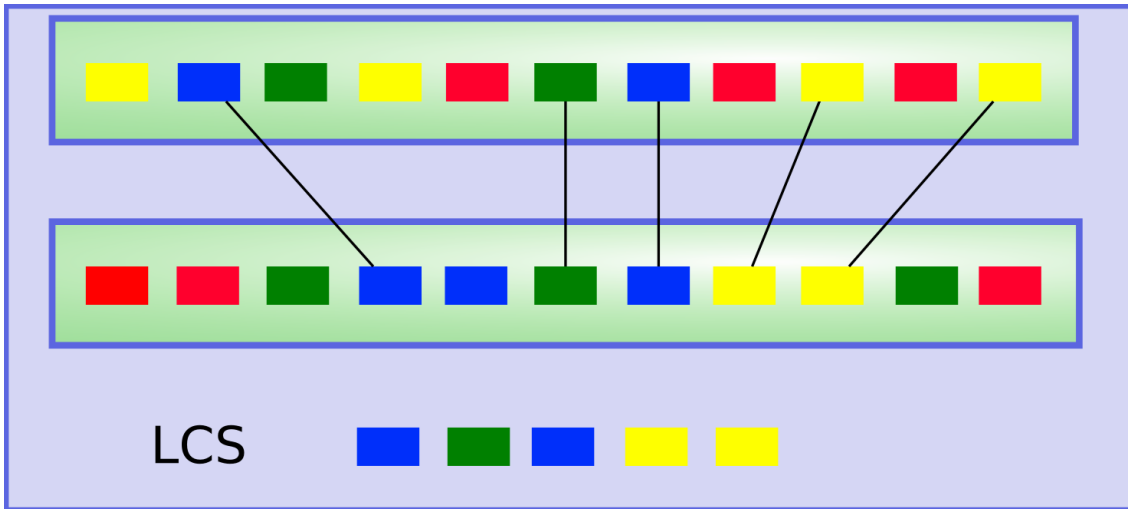
# Algorithme - Ed'Nimbus

- Glisse la fenêtre d'une position

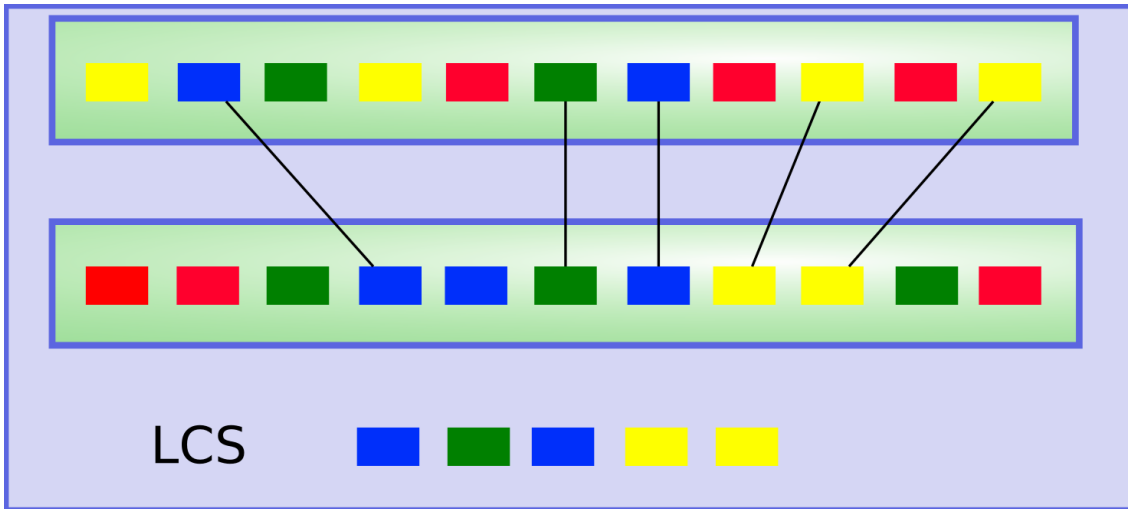


- $r = 3$  (nombre minimum d'occurrences des répétitions)
- $p = 3$  (cond. nécessaire : nb. min. de  $k$ -facteurs partagés)

# 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

Présentation (**biologique**) du problème

Présentation (**algorithmique**) du problème

Nimbus

**Ed'Nimbus**

Algorithme

Quelques résultats

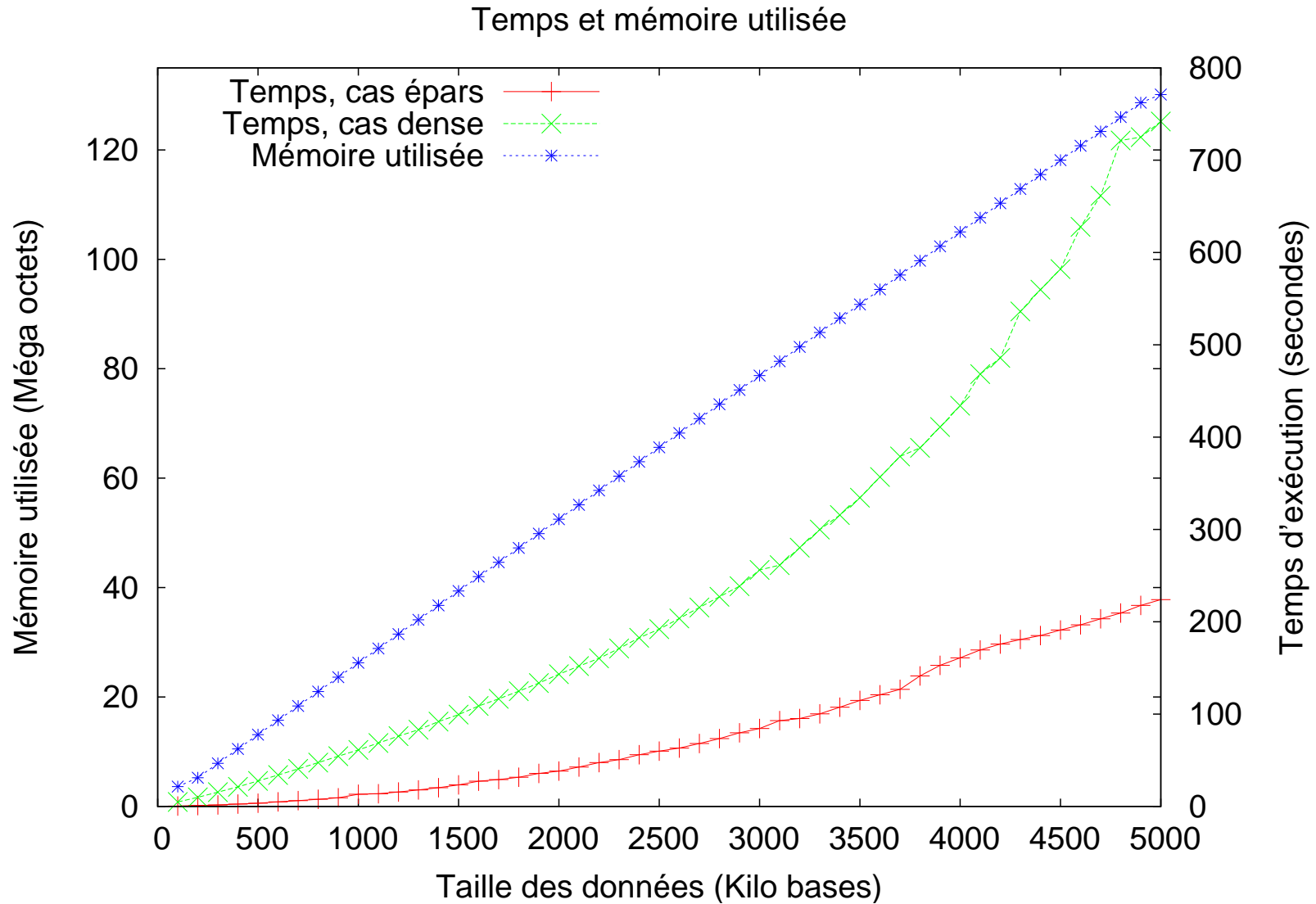
Structures de données utilisées

Arbre des bi-facteurs

Tableau des bi-facteurs

Conclusions

# Mémoire et utilisation du temps



## Estimation de la Spécificité

### Taux de faux positifs

Séquence aléatoire de taille 2 Mb

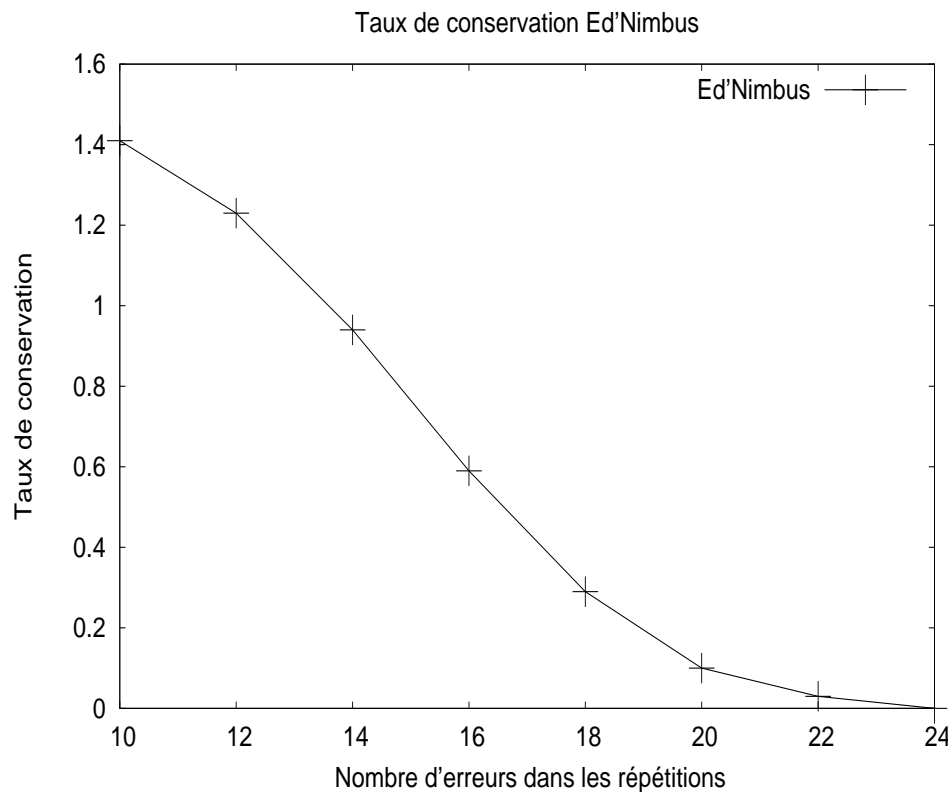
- Contenant 3 répétitions (taille 100, 10 erreurs) :
  - Recherche de 3 répétitions : **0.006 %**
- Contenant 100 répétitions :
  - Recherche de 100 répétitions : **0.34 %**



# Estimation de la spécificité

## Taux de conservation

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



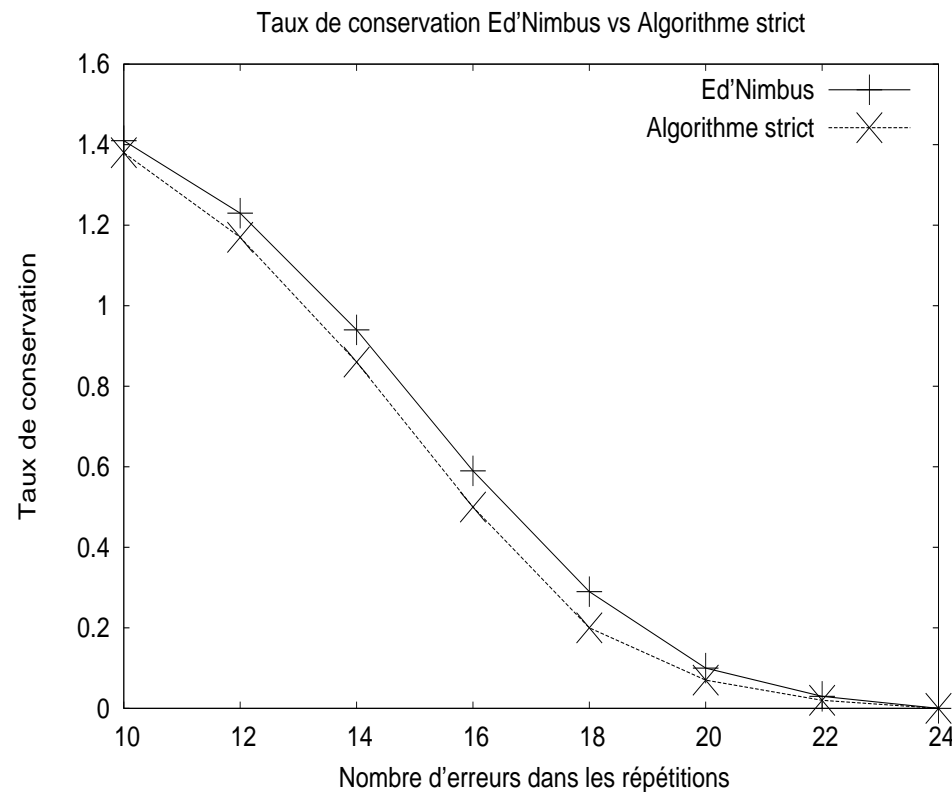
## Données et paramètres

- 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

# Comparaison Ed'Nimbus - «Algorithme stricte» : Spécificité

## Taux de conservation

Nombre de caractères conservés  
-----  
Nombre de caractères appartenant à une répétition

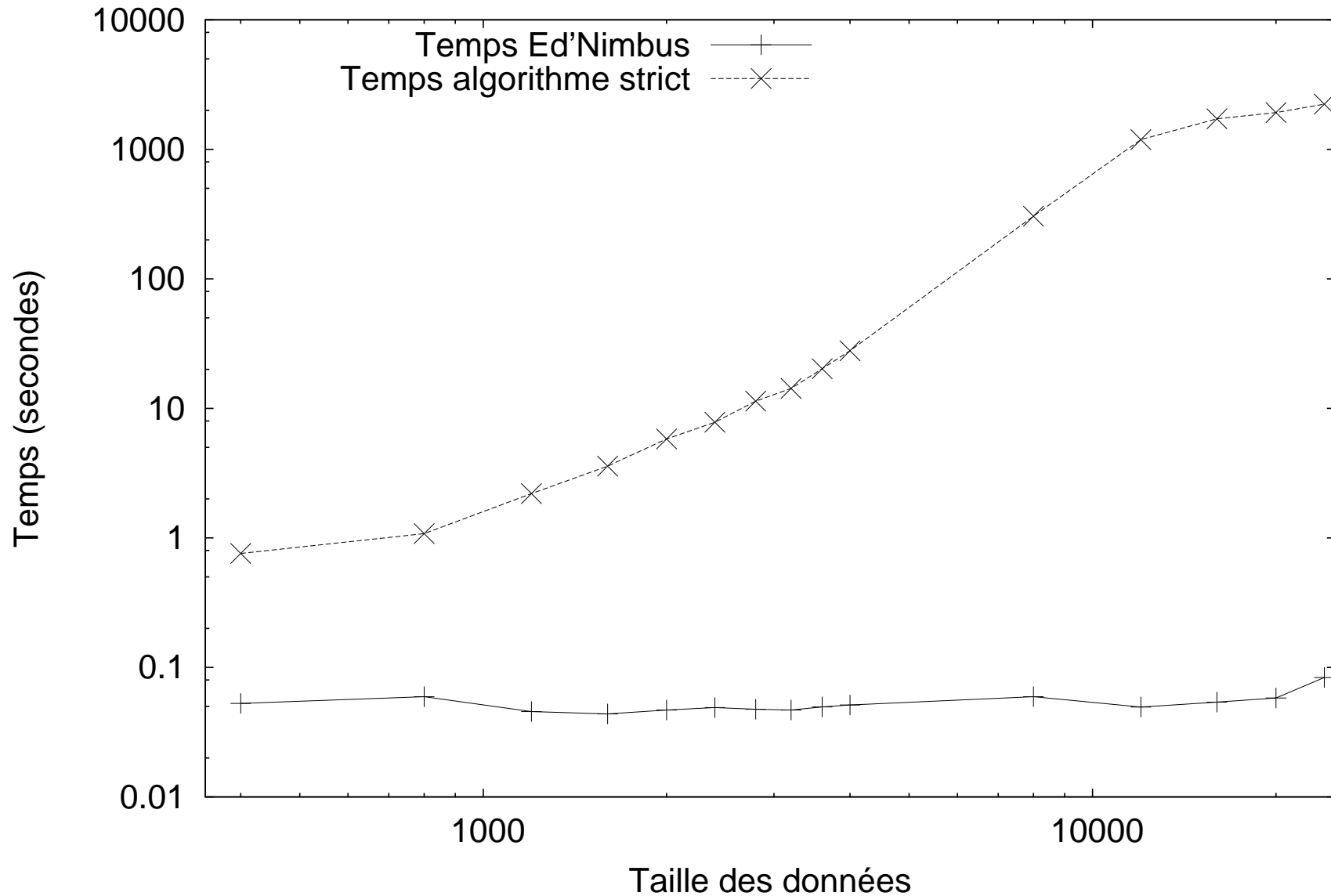


## Données et paramètres

- 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

# Comparaison Ed'Nimbus - «Algorithme stricte» : Temps

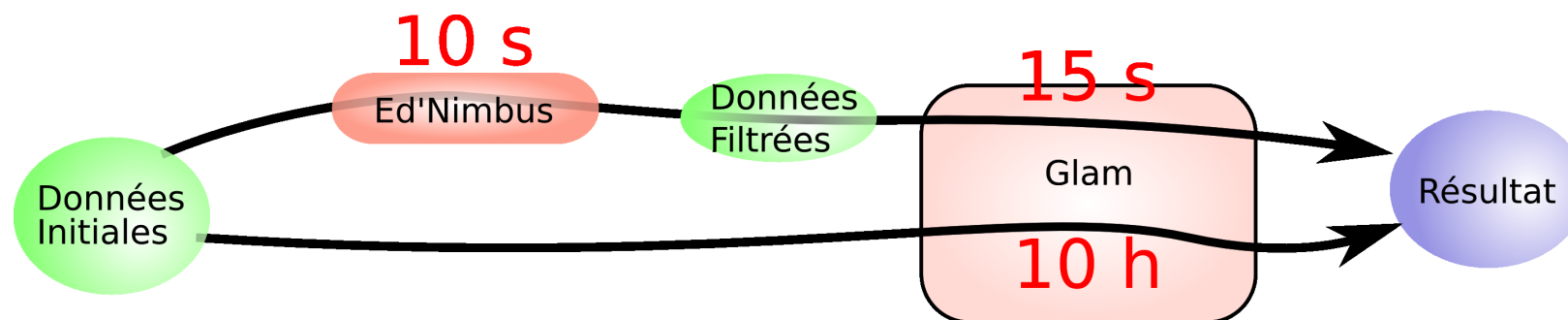
Temps d'exécution, Ed'Nimbus v.s. Algorithme strict



## Accélération d'algorithmes d'alignements locaux

Pré-traitement de séquences (générées) pour accélérer Glam<sup>5</sup>

- 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

<sup>5</sup>Frith, M.; Hansen, U.; Spouge, J.L. & Weng, Z. *Finding Functional Sequence Elements by Multiple Local Alignment*

## Plan de l'exposé

Présentation (**biologique**) du problème

Présentation (**algorithmique**) du problème

Nimbus

Ed'Nimbus

Algorithme

Quelques résultats

**Structures de données utilisées**

Arbre des bi-facteurs

Tableau des bi-facteurs

Conclusions

# Algorithmme - Aperçu

Utilisation de **bi-facteurs**



Structures de données

- Arbre de bi-facteurs
- Tableau des bi-facteurs

## Plan de l'exposé

Présentation (**biologique**) du problème

Présentation (**algorithmique**) du problème

Nimbus

Ed'Nimbus

Algorithme

Quelques résultats

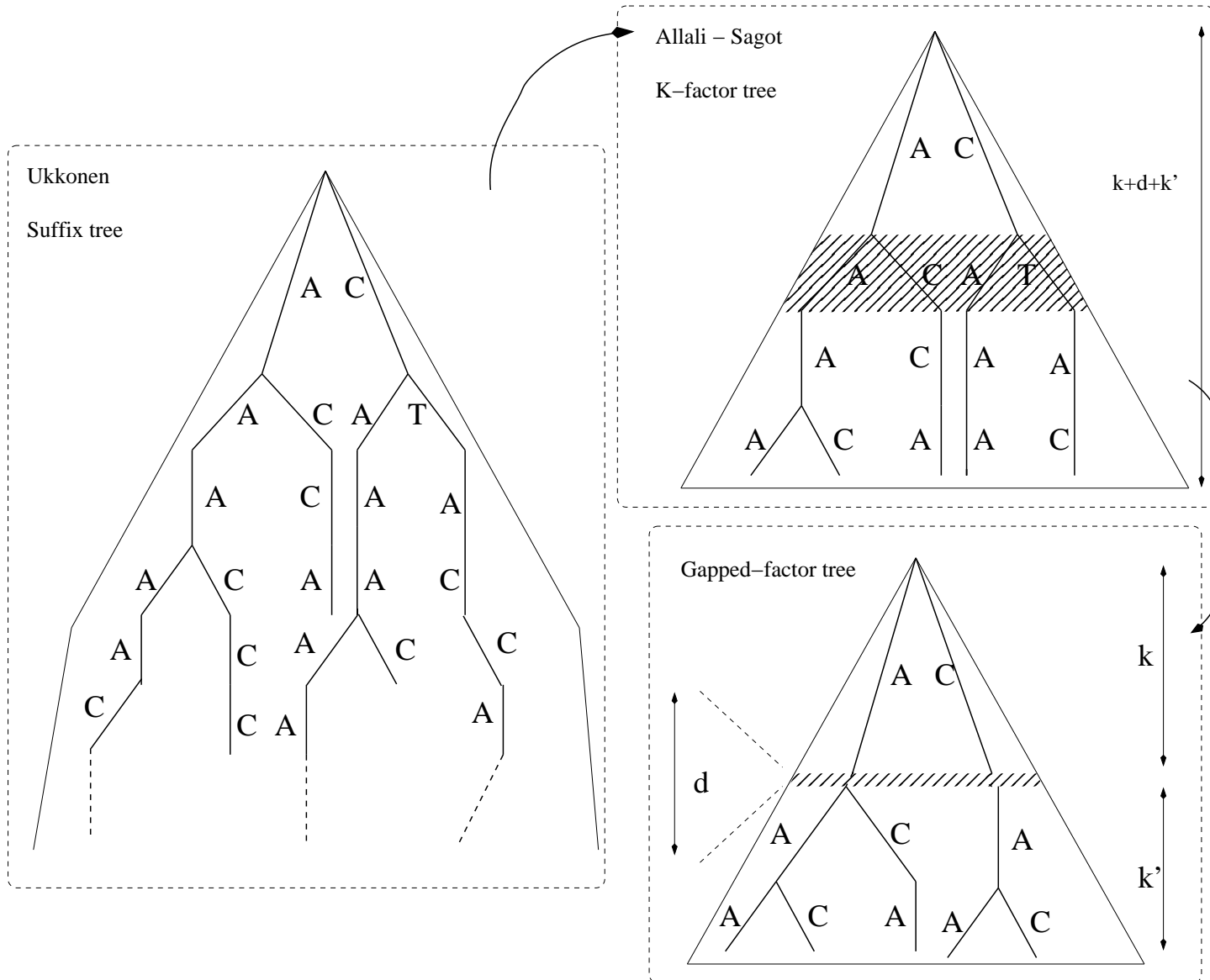
**Structures de données utilisées**

Arbre des bi-facteurs

Tableau des bi-facteurs

Conclusions

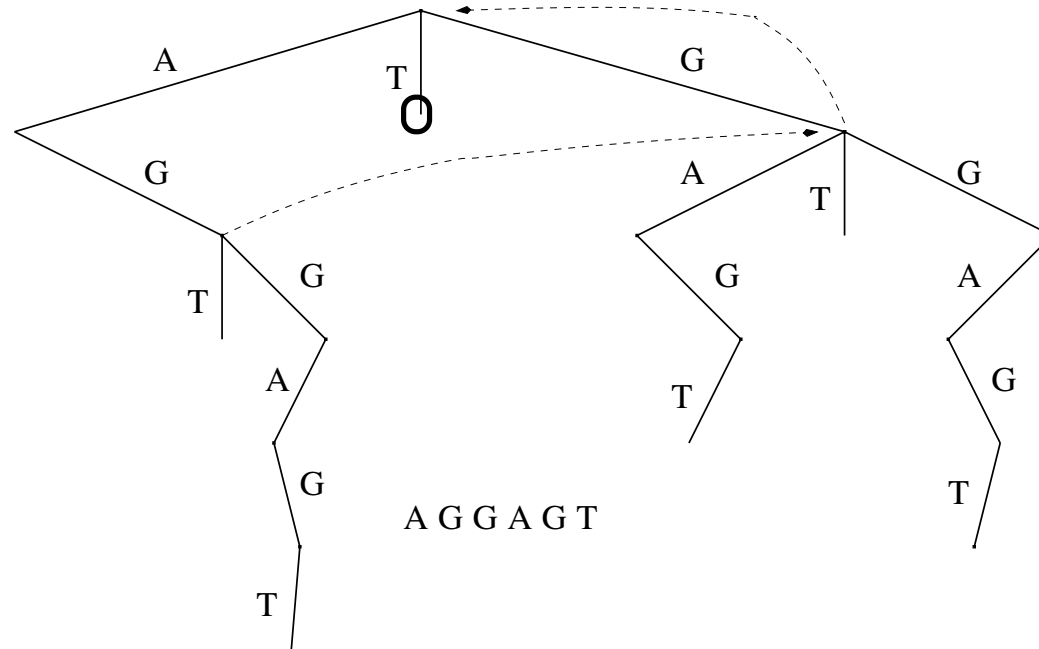
# Idée générale





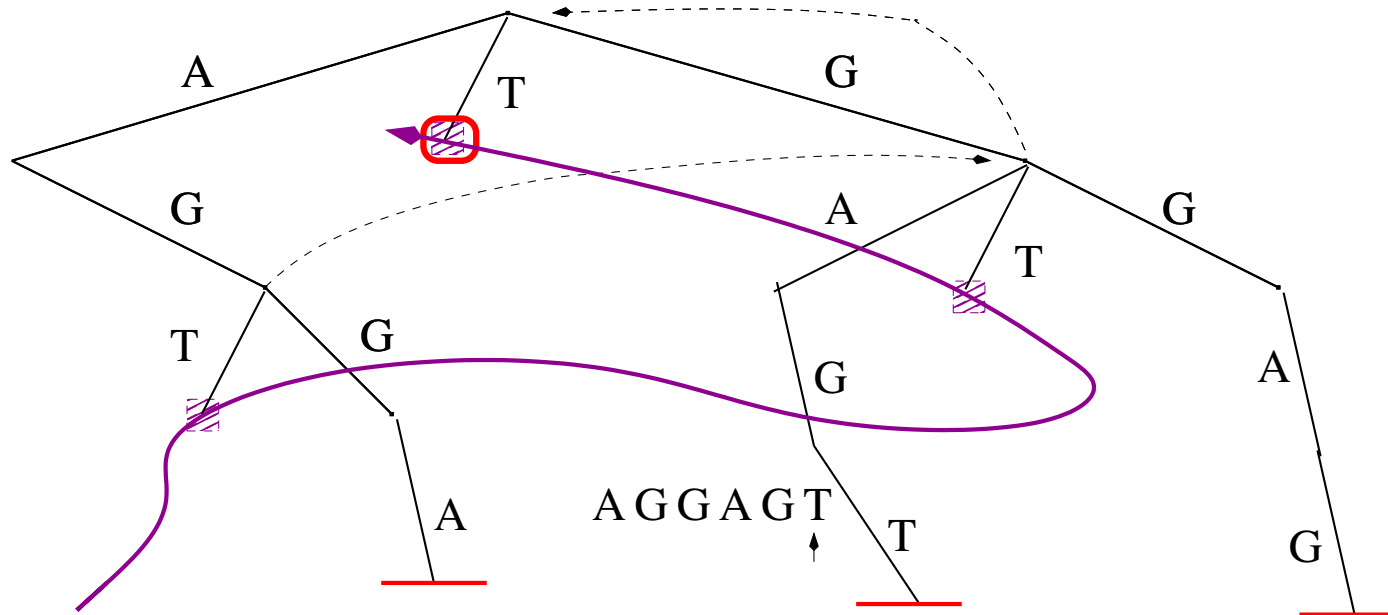
# Aperçu de l'algorithme

Suffix tree (Méthode de construction d'Ukkonen)



# Aperçu de l'algorithme

## *k*-factor tree (Méthode d'Allali-Sagot)





## Plan de l'exposé

Présentation (**biologique**) du problème

Présentation (**algorithmique**) du problème

Nimbus

Ed'Nimbus

Algorithme

Quelques résultats

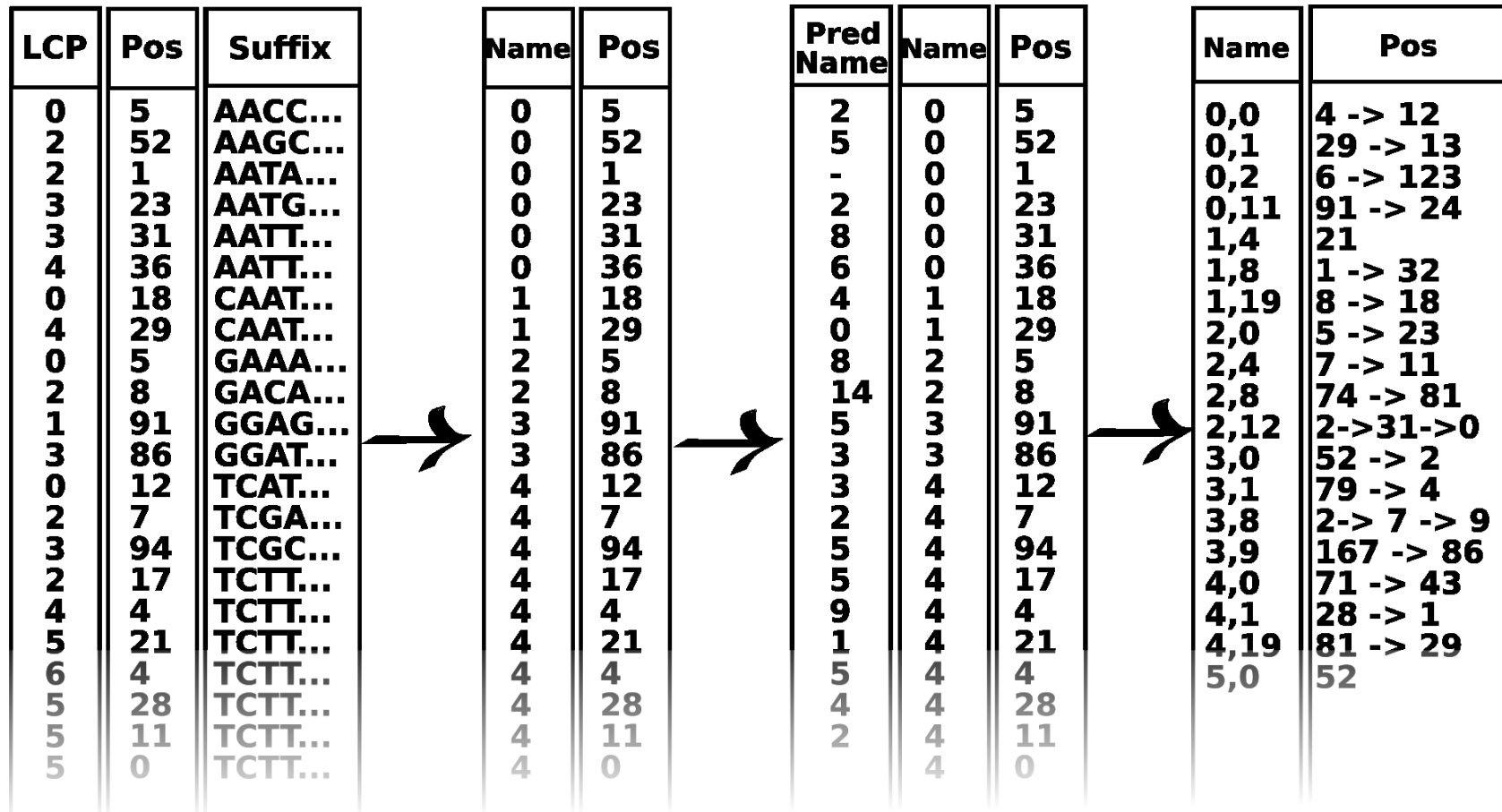
**Structures de données utilisées**

Arbre des bi-facteurs

**Tableau des bi-facteurs**

Conclusions

# Aperçu de l'algorithme



## Plan de l'exposé

Présentation (**biologique**) du problème

Présentation (**algorithmique**) du problème

Nimbus

Ed'Nimbus

Algorithme

Quelques résultats

Structures de données utilisées

Arbre des bi-facteurs

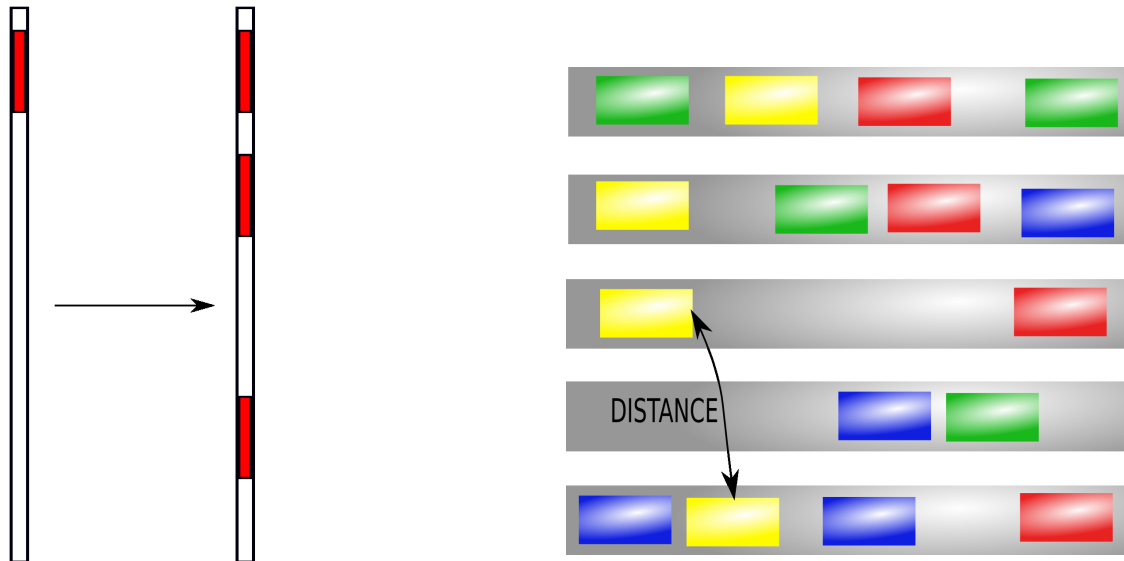
Tableau des bi-facteurs

**Conclusions**

# 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/>