Comparaison de séquences avec amplifications et contractions

Sèverine Bérard et Éric Rivals

Laboratoire d'Informatique, de Robotique
et de Micro-électronique de Montpellier
CNRS - Université Montpellier II
FRANCE

Plan

1. Modèle évolutif.

2. Arches.

3. Algorithme d'alignement.

4. Applications.

Plan

1. Modèle évolutif.
2. Arches.
3. Algorithme d'alignement.
4. Applications.

Modèle Évolutif

- Opérations unitaires
  - **Amplification(A)/Contraction(C)** duplique/supprime un caractère se trouvant à côté d'un caractère identique.

$$a\ b\ c \xrightarrow{\text{Amplification}} a\ b\ b\ c \xrightarrow{\text{Contraction}} a\ b\ c$$

  - **Mutation(M)** mute un caractère en un autre.

$$a\ b\ c \xrightarrow{\text{Mutation}} a\ d\ c$$

  - **Insertion(I)/Délétion(D)** insère/supprime un caractère mais sans contrainte.

$$a\ b\ c \xrightarrow{\text{Insertion}} a\ b\ c\ d \xrightarrow{\text{Délétion}} b\ c\ d$$

- Opérations non unitaires
  - **Génération(G)/Compression(K)** génère/compresse une **arche** à partir/en son caractère ancêtre.

$$a \xrightarrow{\text{Génération}} a\ b\ c\ d\ a \xrightarrow{\text{Compression}} a$$

Exemple d'alignement

Seq1 : A A A B B C B D D B A

Seq2 : A E A A A

Nous voulons obtenir l'alignement optimal suivant :

```
Seq1 : A A A B B C B D D B A
       | [ | [ /   / /   /
Seq2 : A E A A -   - -   -
```

Coût des opérations

- Le modèle est symétrique : I = D et A = C.
- Dû à des observations biologiques : A, C $<$ M, D, I.
- Une délétion peut également être obtenue par une mutation suivie d'une contraction. Nous avons soit :

  1. Hypothèse 1 **(H1)**: D $>$ M + C et I $>$ A + M
  2. Hypothèse 2 **(H2)**: D $\leq$ M + C et I $\leq$ A + M

- **Théorème** : Sous H1 ou H2, le coût de l'alignement est une distance métrique.

Définition du problème

Problème : Soit s et r deux séquences de longueur n et m, trouver l'alignement *global optimal* entre s et r sous le modèle évolutif précédemment défini.

Plan

1. Modèle évolutif.
2. Arches.
3. Algorithme d'alignement.
4. Applications.

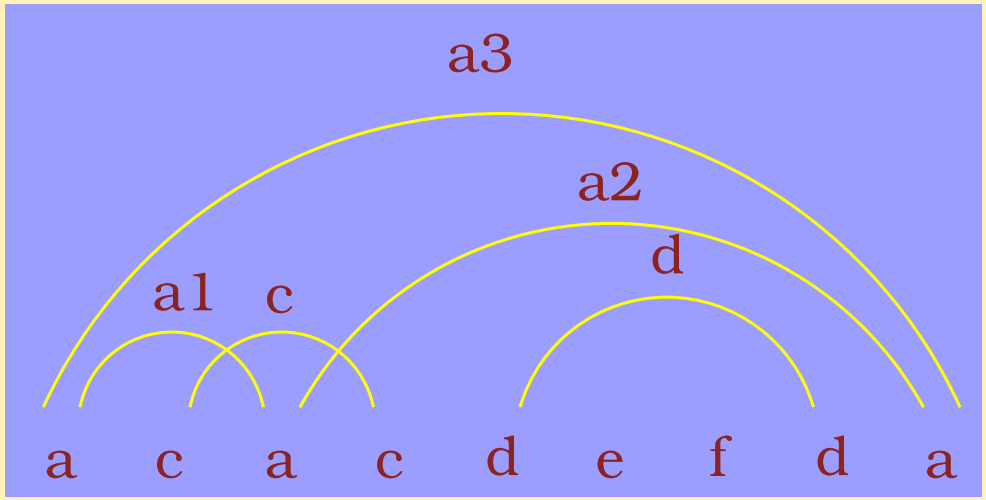
Les arches

Soit s une séquence.

Def : une **arche** de s est un facteur de s dont le premier et le dernier caractère sont identiques.

Def : une arche est **simple** si ses caractères internes apparaissent une seule fois et **complexe** sinon.

Ex: s=acacdefda



Def : deux arches sont **compatibles** si elles ne se croisent pas ou ne partagent pas le même premier ou dernier pied.

Générer/Compresser une arche

- Génération et compression sont symétriques.
- Générer une arche simple : de A à ABCA

Première méthode					Seconde méthode				
	A					A			
Amplification	A	A			Amplification	A	A		
Mutation	A	B			Amplification	A	A	A	
Amplification	A	B	B		Amplification	A	A	A	A
Mutation	A	B	C		Mutation	A	B	A	A
Amplification	A	B	C	C	Mutation	A	B	C	A
Mutation	A	B	C	A					

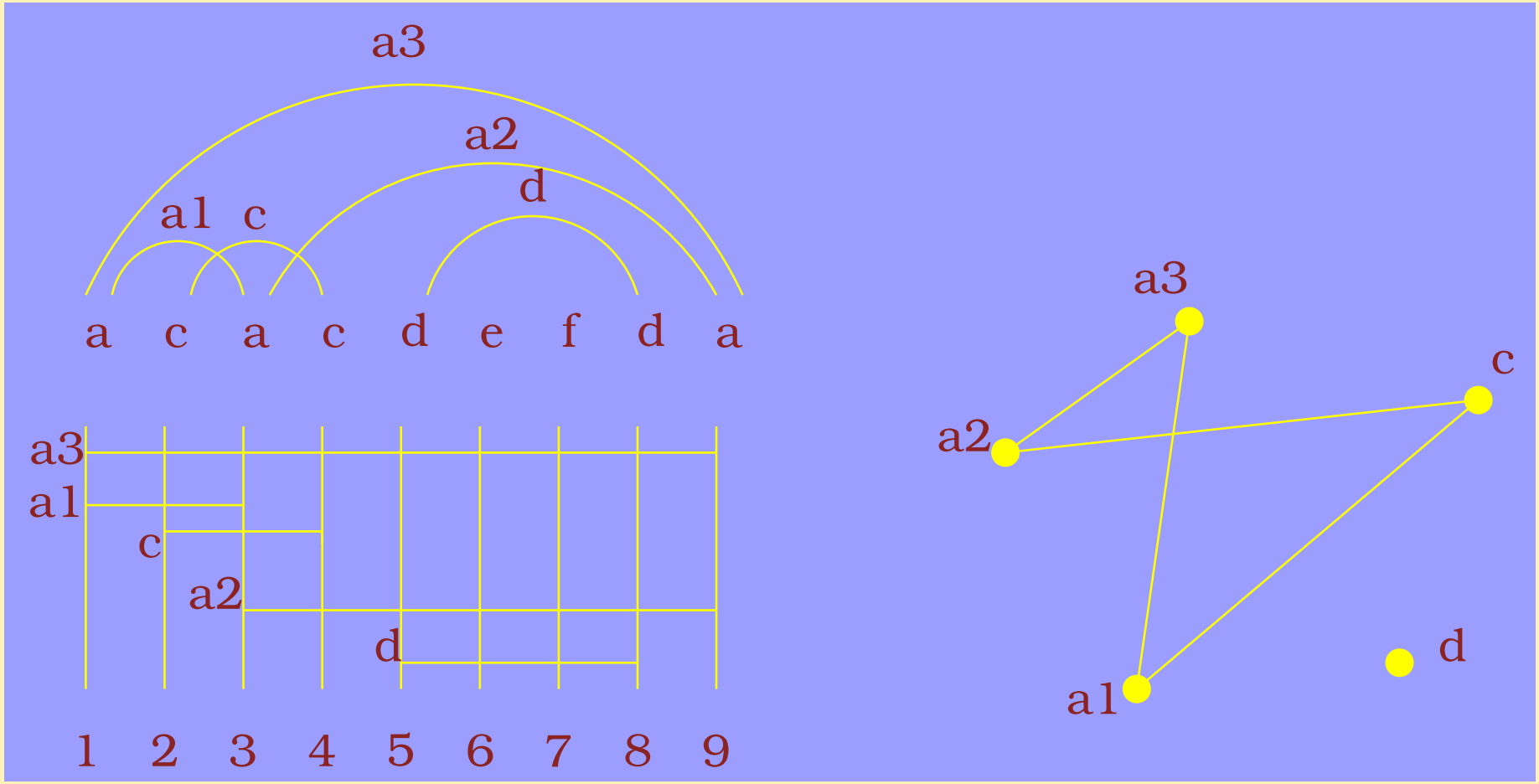
Les caractères identiques aux extrémités permettent d'éviter une mutation.

- Les opérations ne sont pas commutatives.
- **Théorème :** la génération/compression optimale d'une arche utilise le plus grand sous-ensemble d'arches internes compatibles.

Graphes de recouvrement (Overlap graphs)

- Séquence de longueur n, les arches sont des sous-intervalles de [1,n].
- La relation de compatibilité défini un graphe de recouvrement.
- **Théorème**: Trouver le plus grand ensemble d'arches deux à deux compatibles est équivalent à trouver le stable max dans un graphe de recouvrement.
- La complexité de cette procédure est $O(V^2)$ avec V le nombre de sommets du graphe [Apostolico et al. 92].
- Le nombre maximal d'arches est $O(n^2)$.

Exemple



Comment convertir le problème de l'ensemble maximum d'arches compatibles en problème de stable max dans un graphe de recouvrement.

Plan

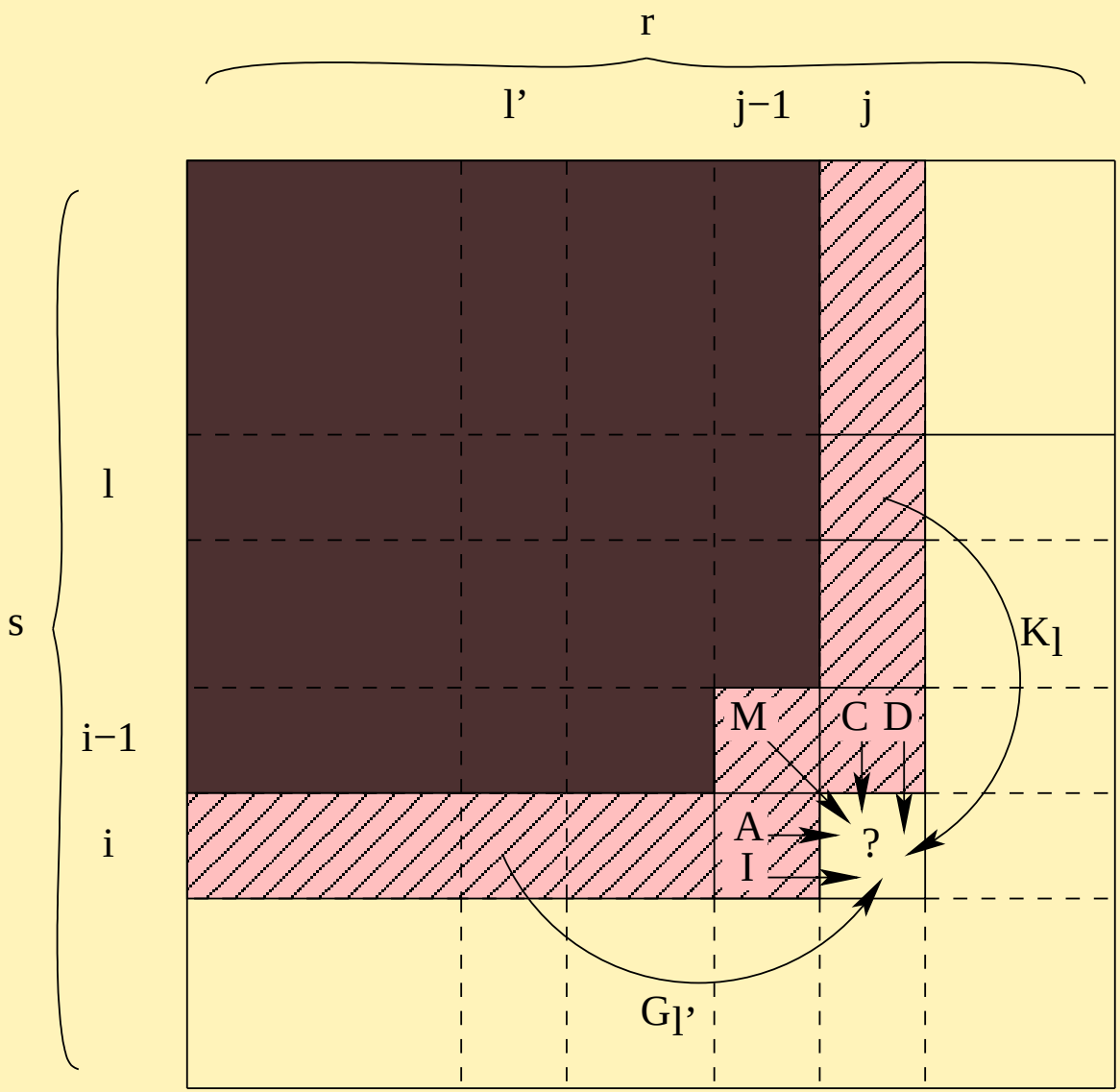
1. Modèle évolutif.

2. Arches.

3. Algorithme d'alignement.

4. Applications.

Matrice de programmation dynamique



Récurrence

$$\mathscr{A}[i,j] = min \begin{cases} \mathscr{A}[i-1, j-1] + M(s[i],r[j]) & \text{Mutation} \\ \mathscr{A}[i-1, j] + C \text{ } \textbf{iff s[i - 1] = s[i]} & \text{Contraction} \\ \mathscr{A}[i-1, j] + D & \text{Délétion} \\ \mathscr{A}[i, j-1] + A \text{ } \textbf{iff r[j - 1] = r[j]} & \text{Amplification} \\ \mathscr{A}[i, j-1] + I & \text{Insertion} \\ \mathscr{A}[l, j] + K(s[l, i]) & \text{Compression d'arche} \\ \quad \forall \textbf{ l} \in \textbf{[1, i - 2] tel que s[l] = s[i]} & \\ \mathscr{A}[i, l'] + G(r[l',j]) & \text{Génération d'arche} \\ \quad \forall \textbf{ l'} \in \textbf{[1, j - 2] tel que r[l'] = r[j]} & \end{cases}$$

Complexité

Théorème : Si p = max(n, m), la complexité en temps globale est $O(p^4)$.

Plan

1. Modèle évolutif.

2. Arches.

3. Algorithme d'alignement.

4. Applications.

Les minisatellites (ms)

- Séquences d'ADN appartenant à la classe des séquences répétées en tandem :

  $\cdots$ cggcgat cggcgac cggagat cggcgat cggcgat cggagat cgacgat $\cdots$

- ms évoluent principalement grâce aux amplifications en tandem et contractions en tandem

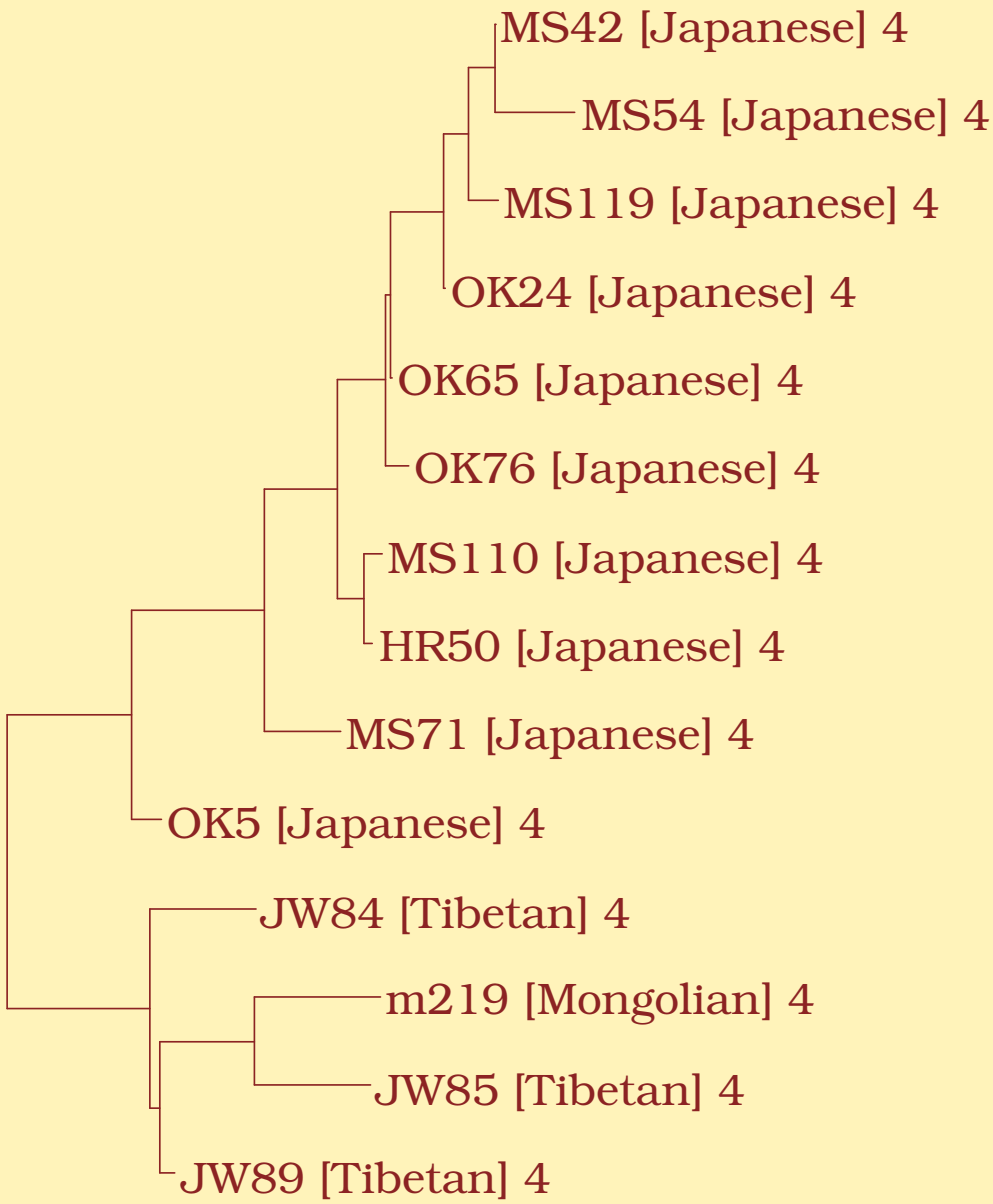
- Les minisatellites sont difficiles à séquencer $\Rightarrow$ une méthode spécifique : **Minisatellite Variant Repeat PCR** [Jeffreys et al. 91].
  MVR-PCR fournit une **carte de ms**:

  - Nouvel alphabet : A = cggcgat B = cggcgac C = cggagat D = cgacgat

  - Carte correspondante : A B C A A C D

- Expériences sur un jeu de 690 cartes du minisatellite humain MSY1 : Calcul de tous les alignements deux à deux, construction de l'arbre phylogénétique avec BioNJ à partir de notre matrice de distance.

Exemple d'arbre phylogénétique



Arbre phylogénétique de l'haplogroupe 4.

Conclusion et perspectives

- Une nouvelle méthode d'alignement prenant en considération les opérations d'amplification et de contraction

- Extensions du modèle: amplifications/contractions multiples

  1. arité > 1: $\cdots$ B $\underline{\text{D}}$ A $\cdots \longrightarrow \cdots$ B **D D D** A $\cdots$

  2. ordre > 1: $\cdots$ B $\underline{\text{D A}}$ $\cdots \longrightarrow \cdots$ B **D A D A** $\cdots$

- Application pour d'autres problèmes biologiques et différents minisatellites