

Analyse de séquences

Partie II-1 : Dot Plot et Prog. dyn.

Sèverine Bérard



AMAP - Université Montpellier 2



-
- Introduction
 - Dotplot
 - Alignement : définitions et scores
 - Algorithme d'alignement (programmation dynamique)
 - Autres types d'alignements
 - La suite ...

Alignement = comparaison

- Recherche d'homologie
- Prédiction de gènes
- Recherche de fonction commune / transfert d'annotation

On dispose de beaucoup de séquences \Rightarrow ressource à exploiter

Score = 271, Expect = 2e-24

```
B FVNQHLCGSHLVEALYLVCGERGFFYTPKAGIVEQCCASTCSLYQLENYCN
  |||...|||
É FVNQHLCGSHLVEALYLVCGERGFFYTPKTGIVEQCCTGVCSLYQLENYCN
```

Score = 256, Expect = 1e-22

```
B NQHLCGSHLVEALYLVCGERGFFYTPKAGIVEQCCASTCSLYQLENYCN
  |||...|||+||
C NQHLCGSHLVEALYLVCGERGFFYSPKTGIVEQCCENPCSLYQLENYCN
```

Score = 243, Expect = 4e-21

```
B FVNQHLCGSHLVEALYLVCGERGFFYTPKAGIVEQCCASTCSLYQLENYCN
  |...|+||+| |+ |...|+|| |...|+|||
O FPNQHLCGSHLVEALYLVCGEKGFYYIPRMGIVEECCKGVCSMYQLENYCN
```

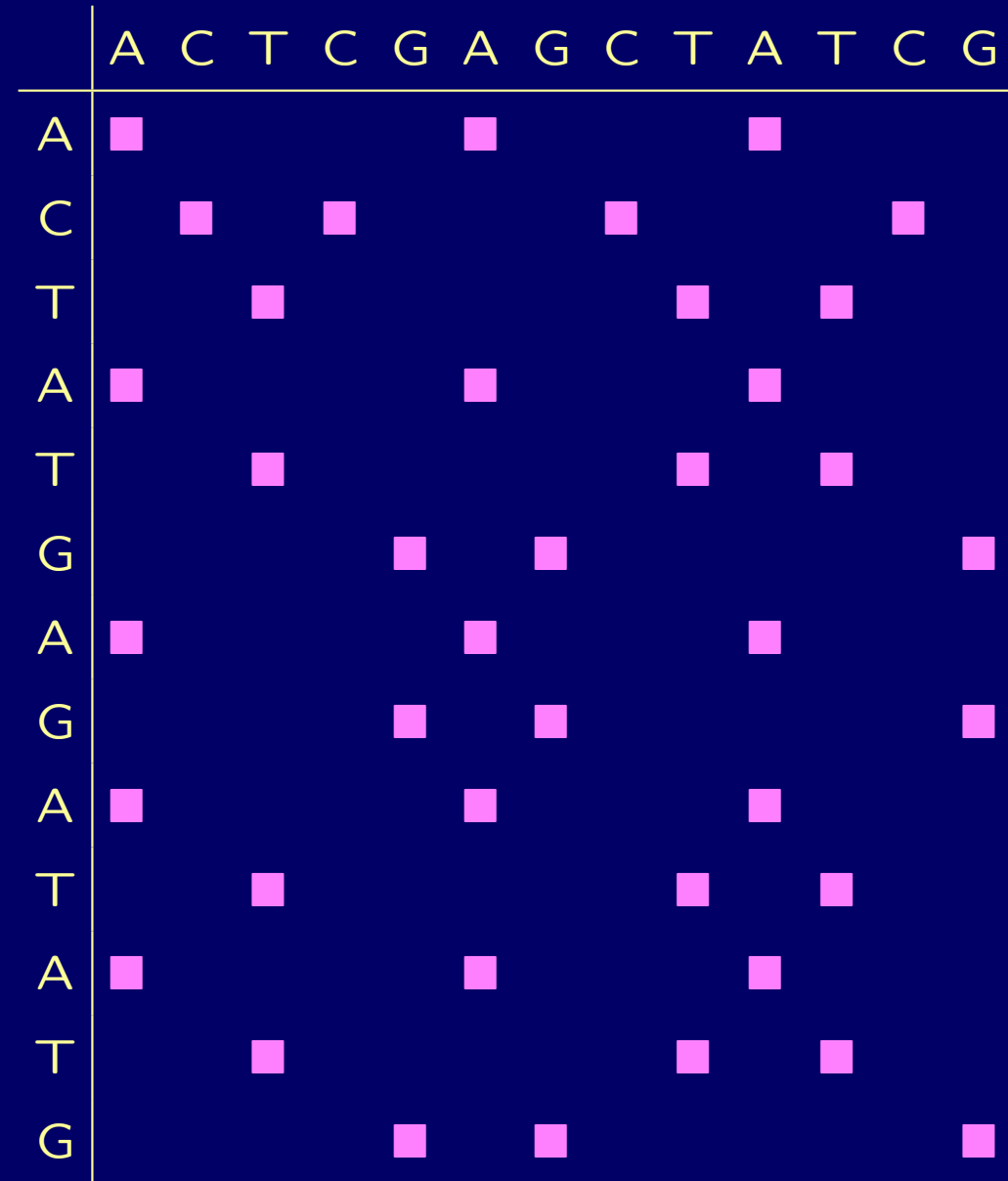
Insuline chez la Baleine, l'Éléphant, le Canard et l'Ornithorynque (EMBL+BLASTP)

-
- Introduction
 - **Dotplot**
 - Alignement : définitions et scores
 - Algorithme d'alignement (programmation dynamique)
 - Autres types d'alignements
 - La suite ...

match (identité) → ■

mismatch → □

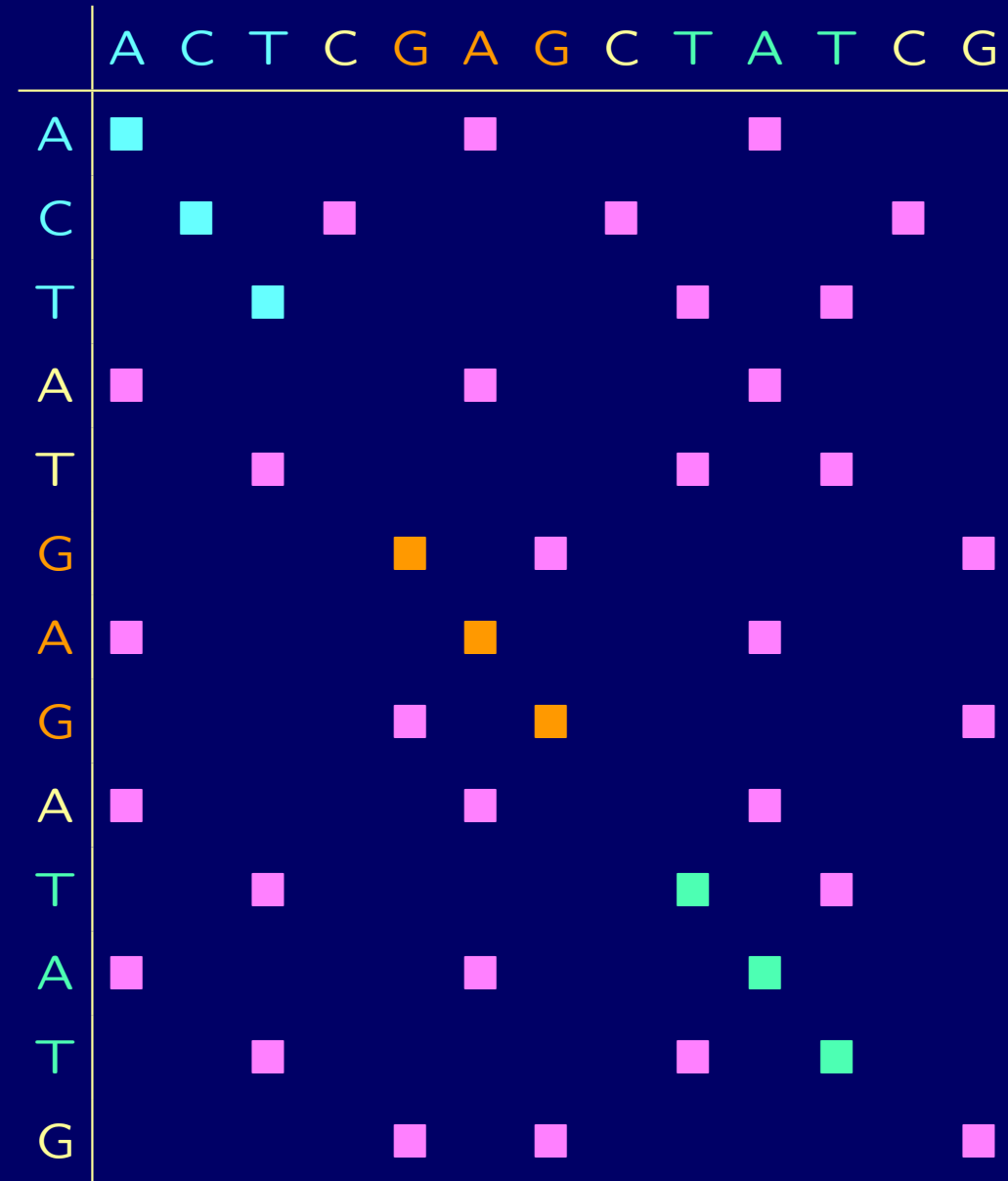
diagonale = région similaire



match (identité) → ■

mismatch → □

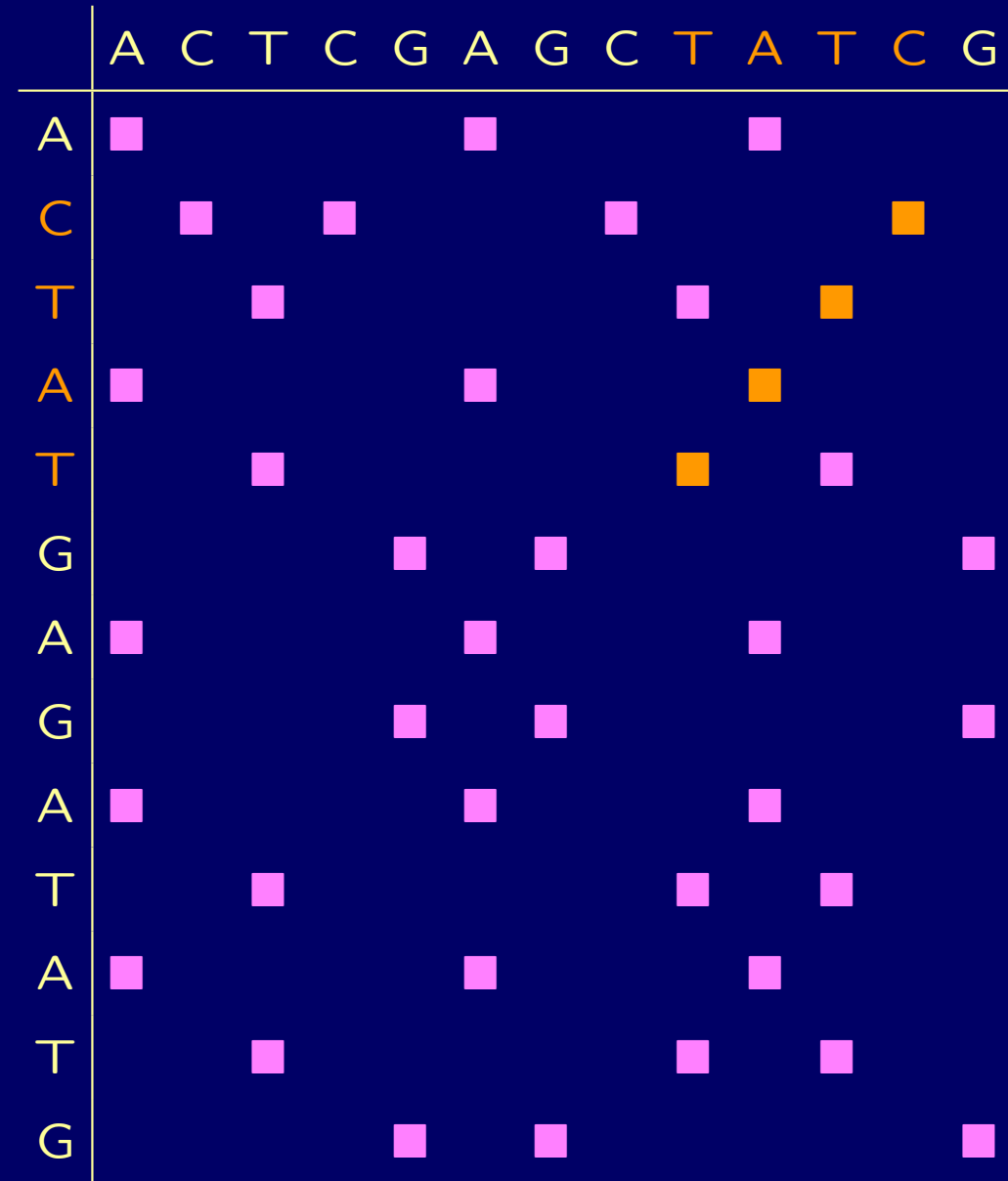
diagonale = région similaire



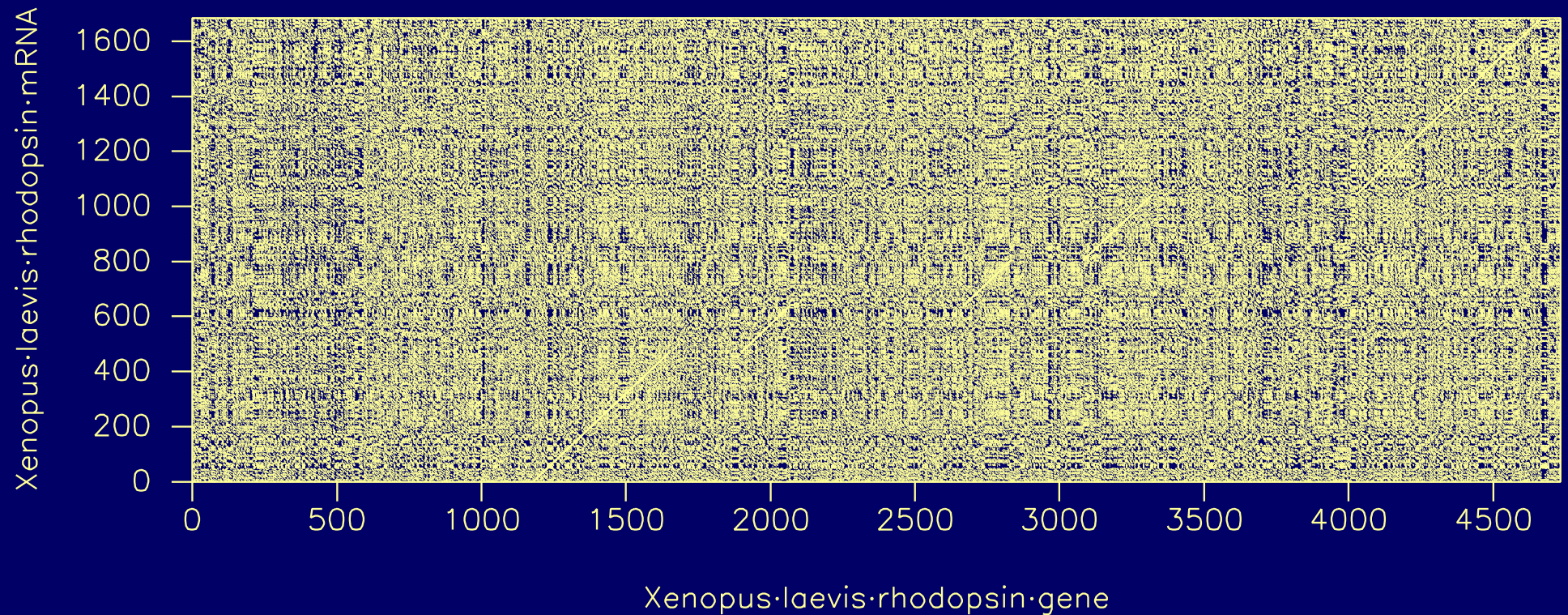
match (identité) → ■

mismatch → □

diagonale inversée = région miroir

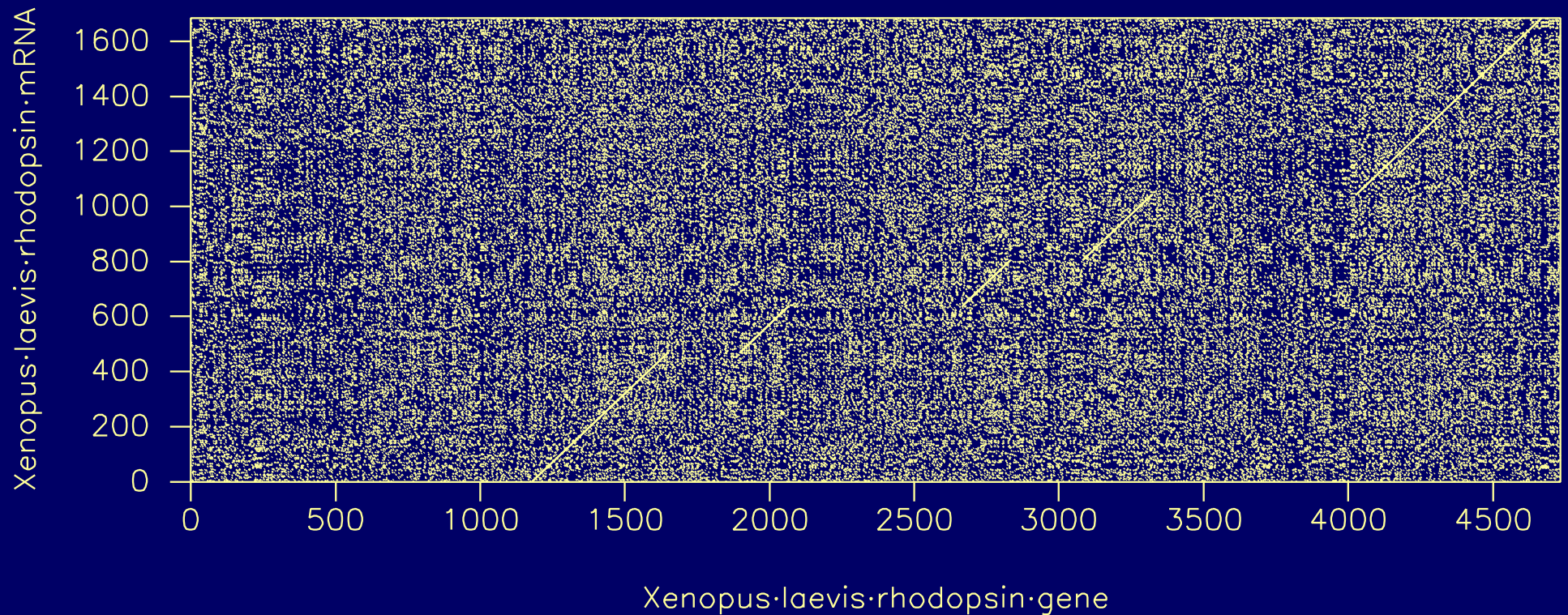


dotpath (08/06/05)



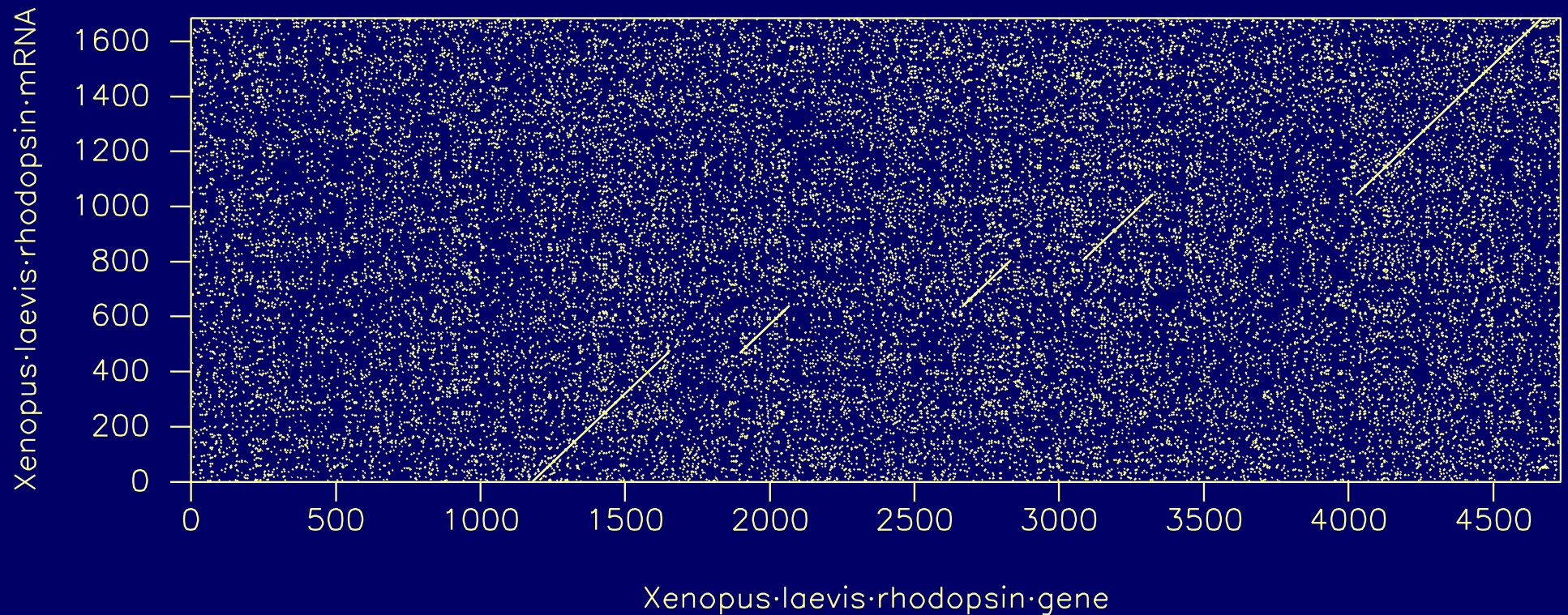
Taille fenêtre = 2

dotpath (08/06/05)



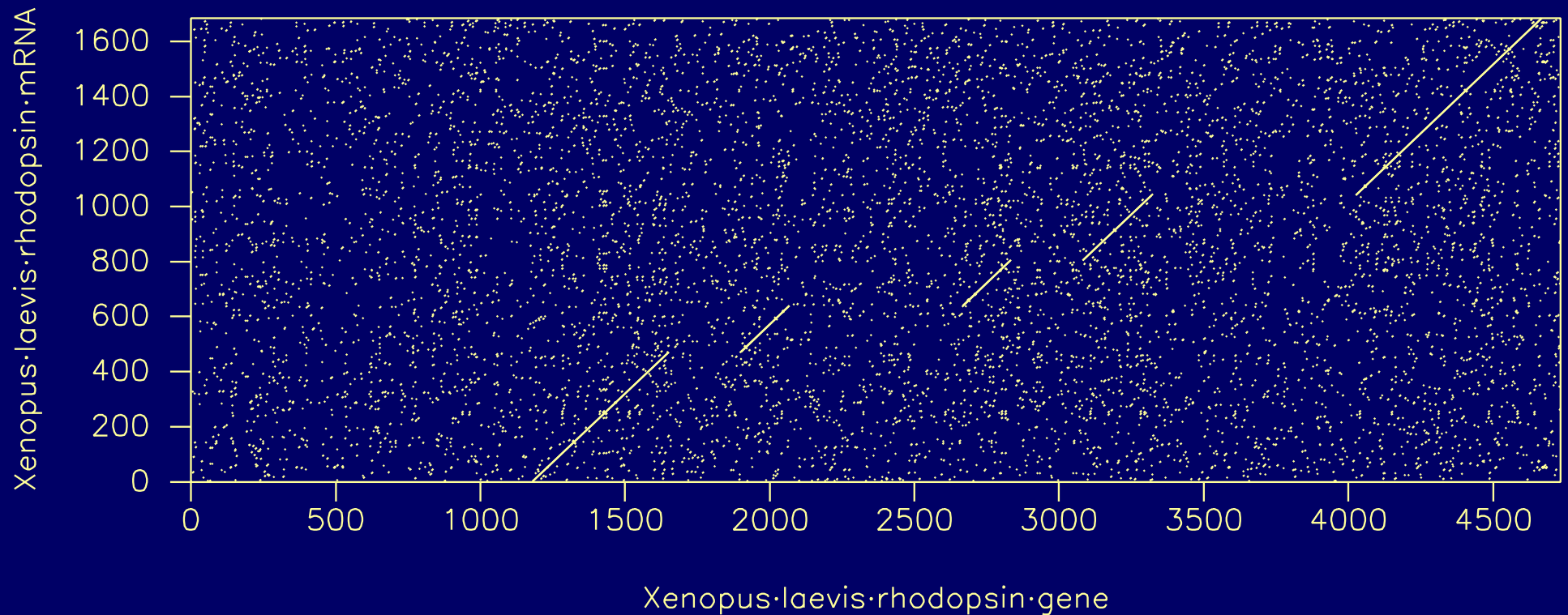
Taille fenêtre = 3

dotpath (08/06/05)



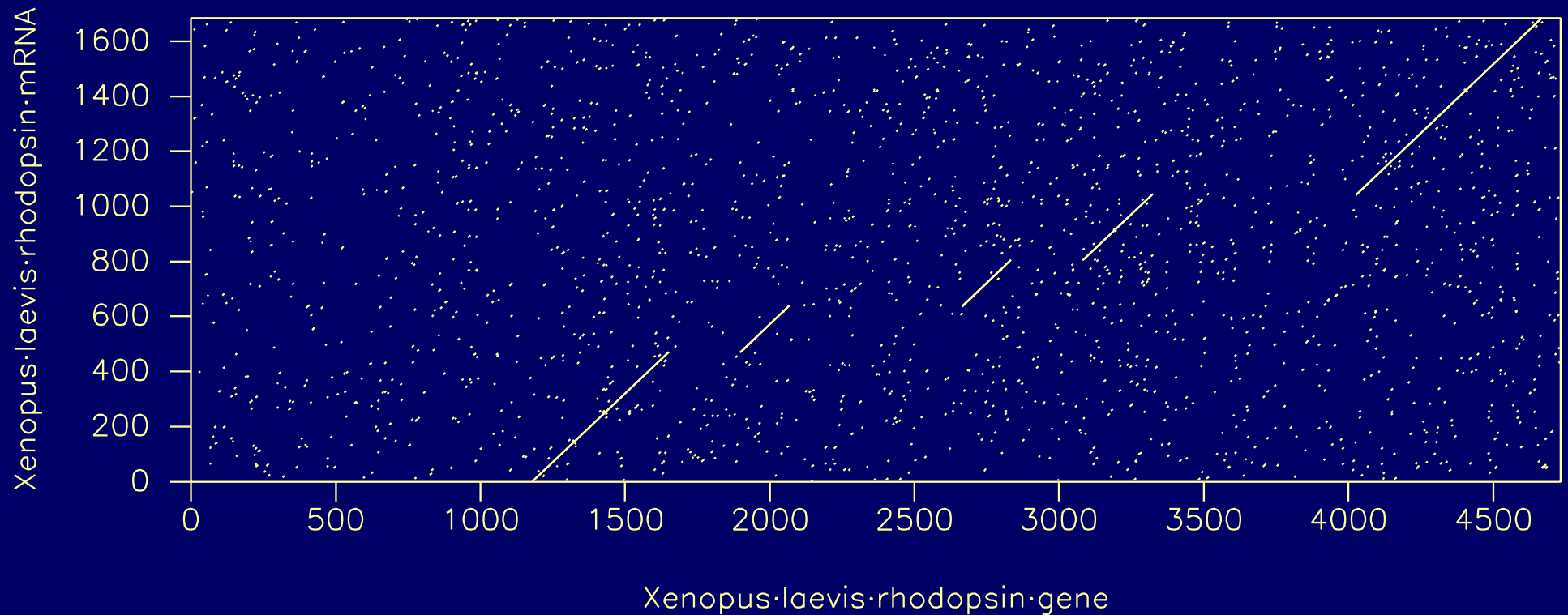
Taille fenêtre = 4

dotpath (08/06/05)



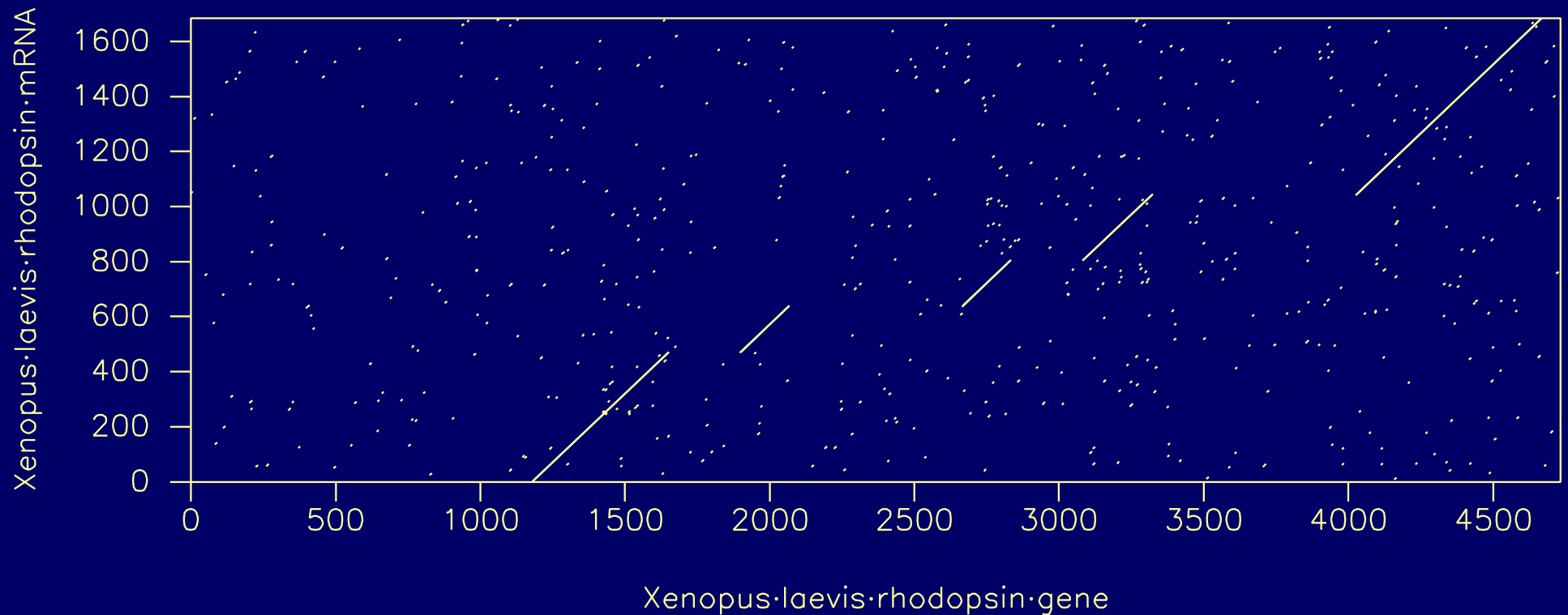
Taille fenêtre = 5

dotpath (08/06/05)



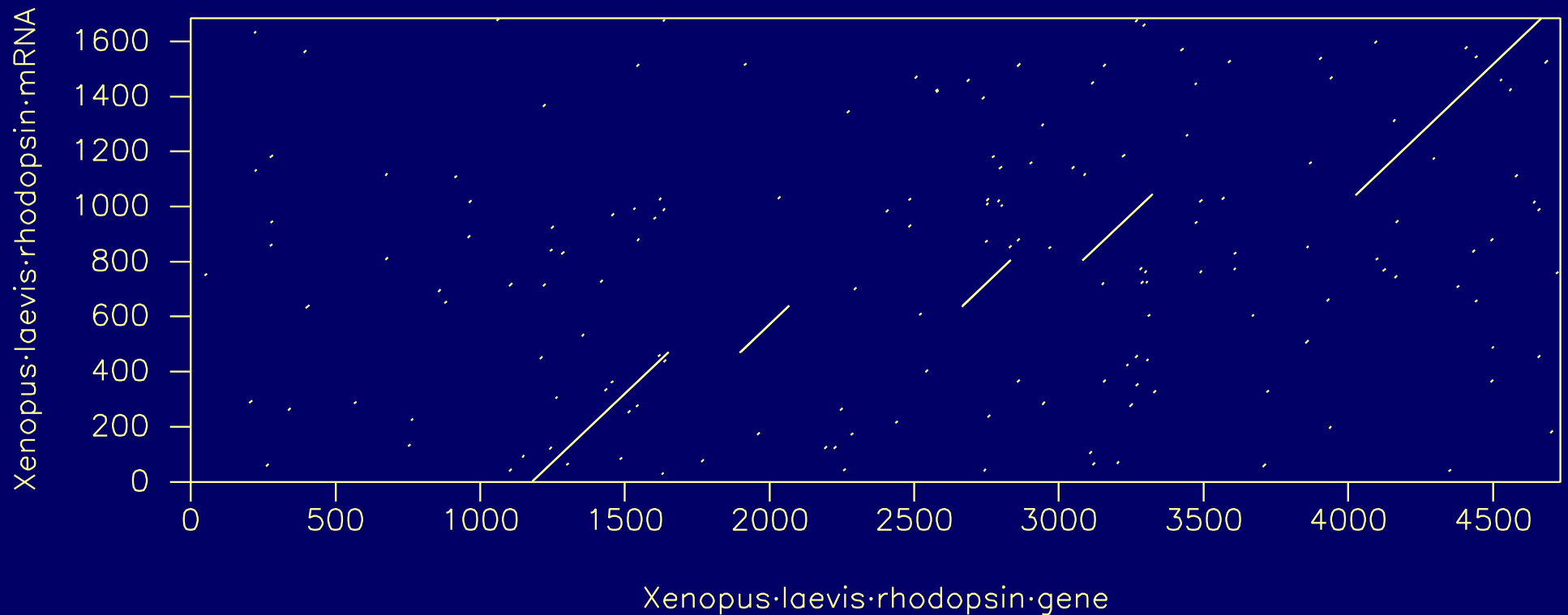
Taille fenêtre = 6

dotpath (08/06/05)



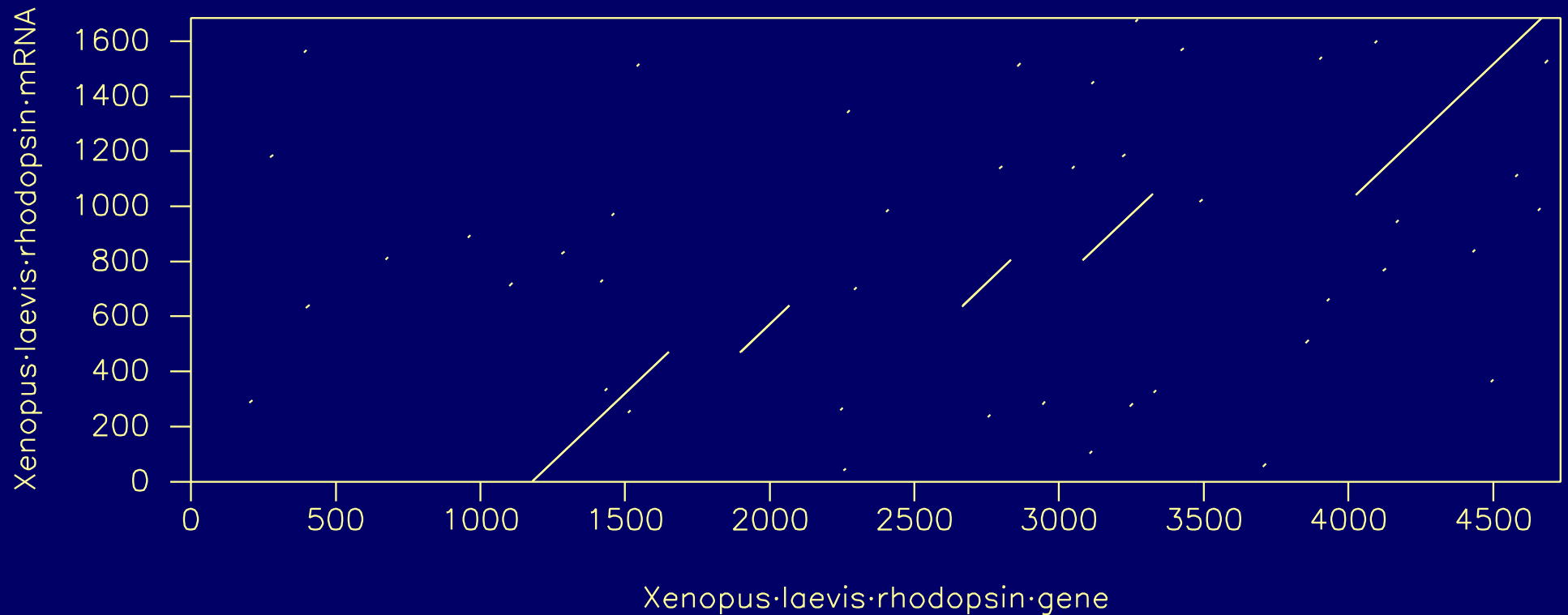
Taille fenêtre = 7

dotpath (08/06/05)



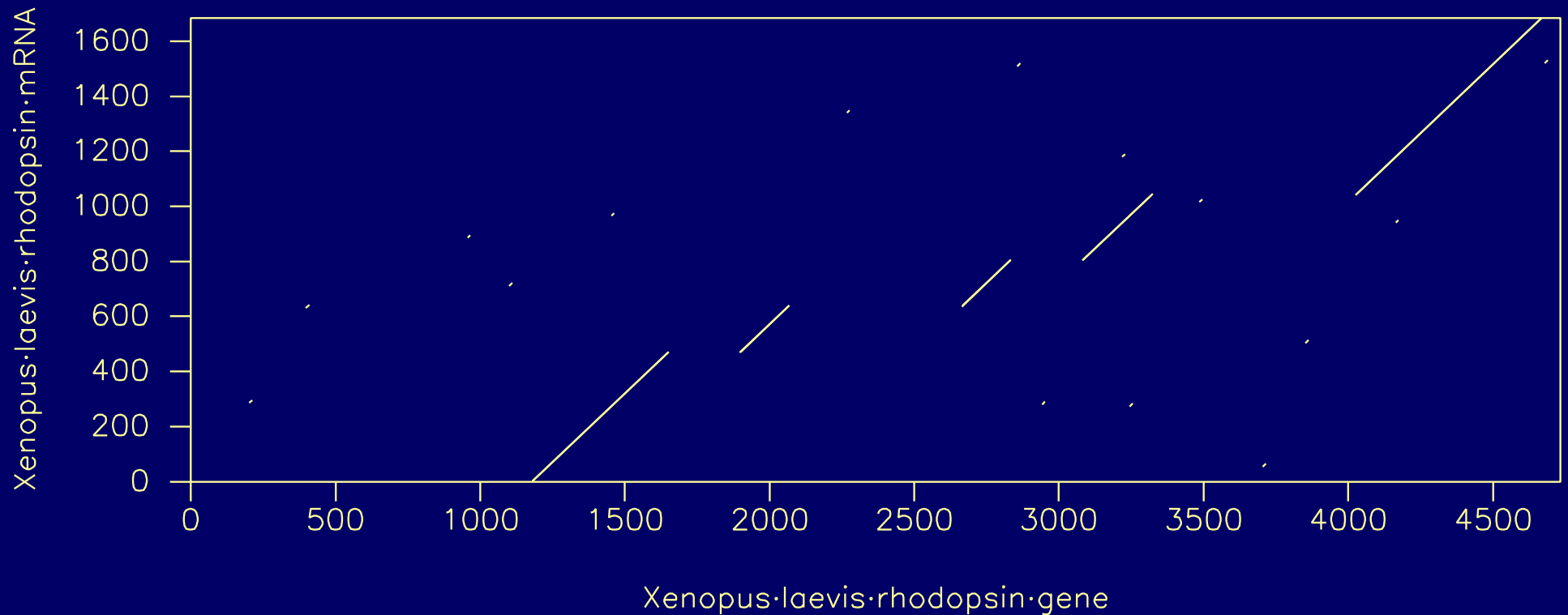
Taille fenêtre = 8

dotpath (08/06/05)



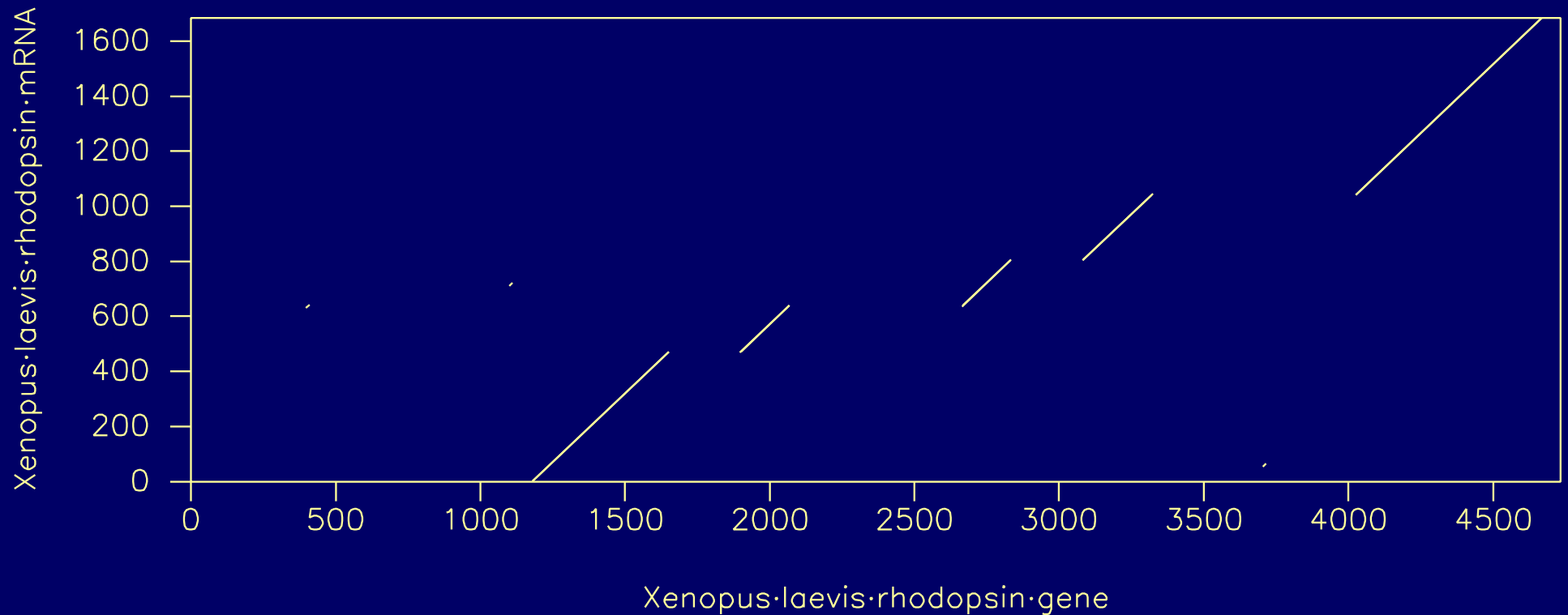
Taille fenêtre = 9

dotpath (08/06/05)



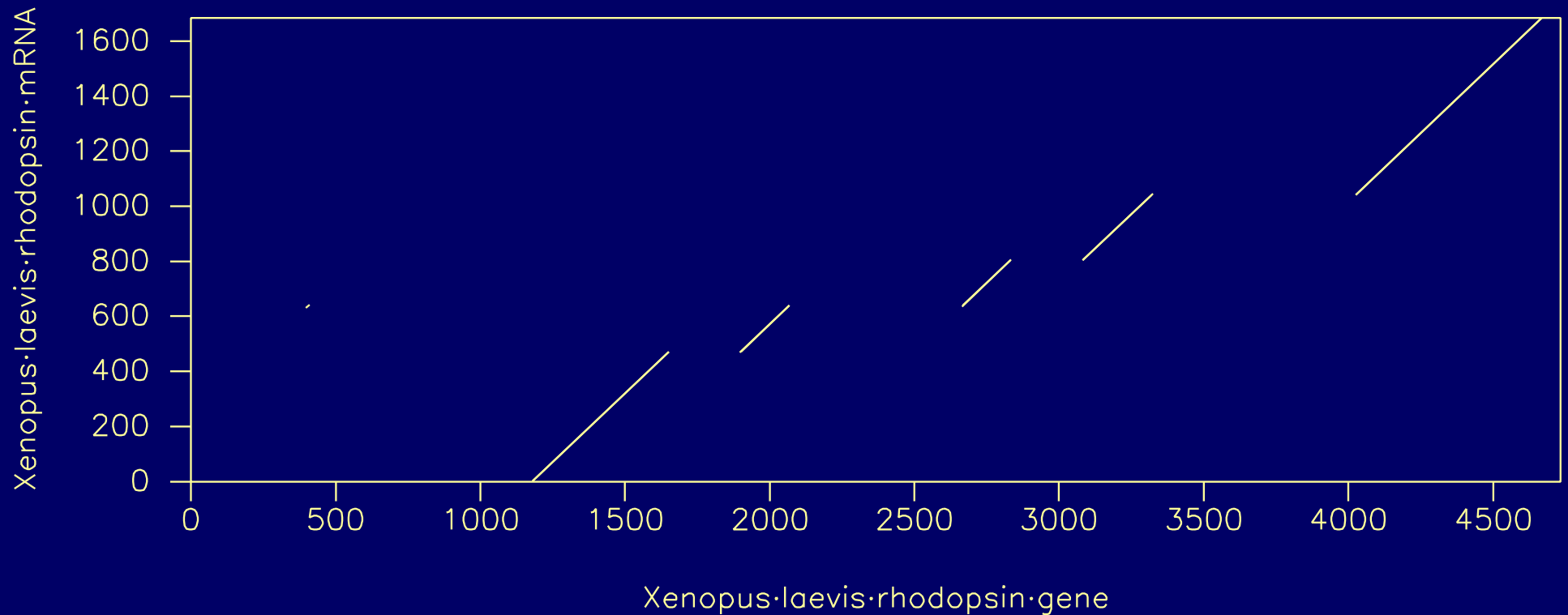
Taille fenêtre = 10

dotpath (08/06/05)



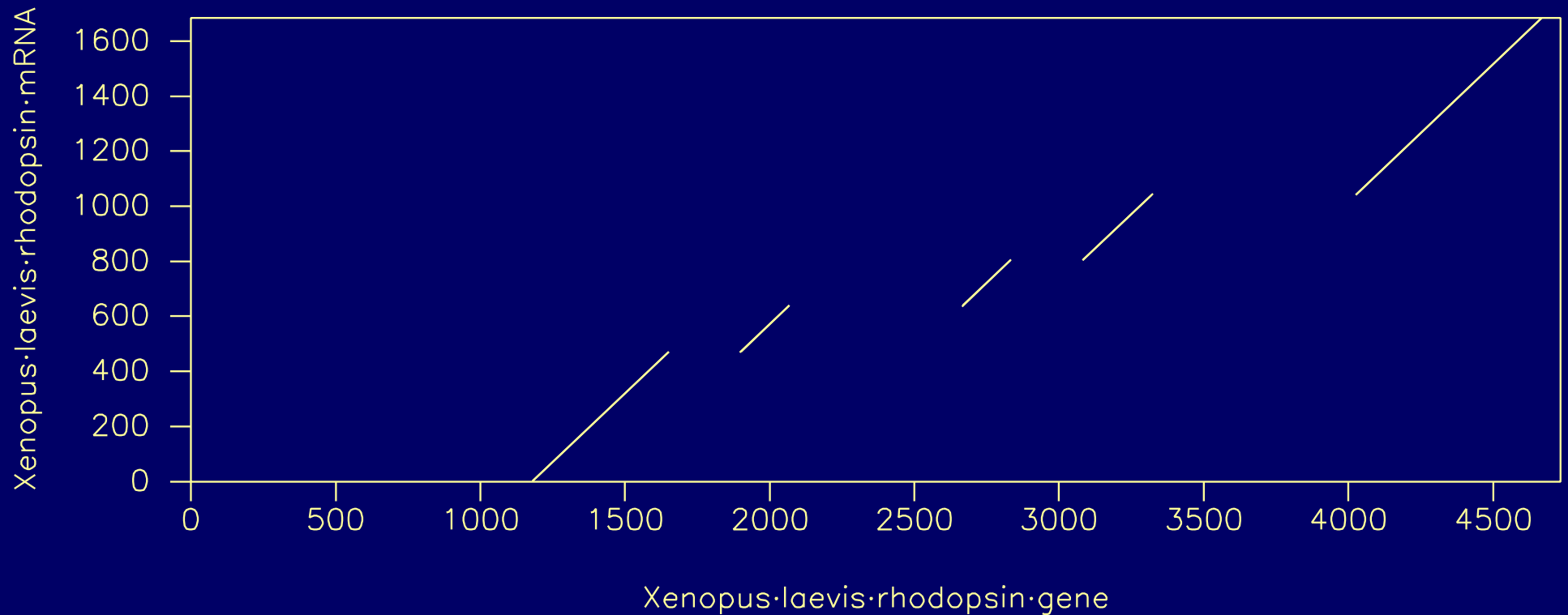
Taille fenêtre = 11

dotpath (08/06/05)



Taille fenêtre = 12

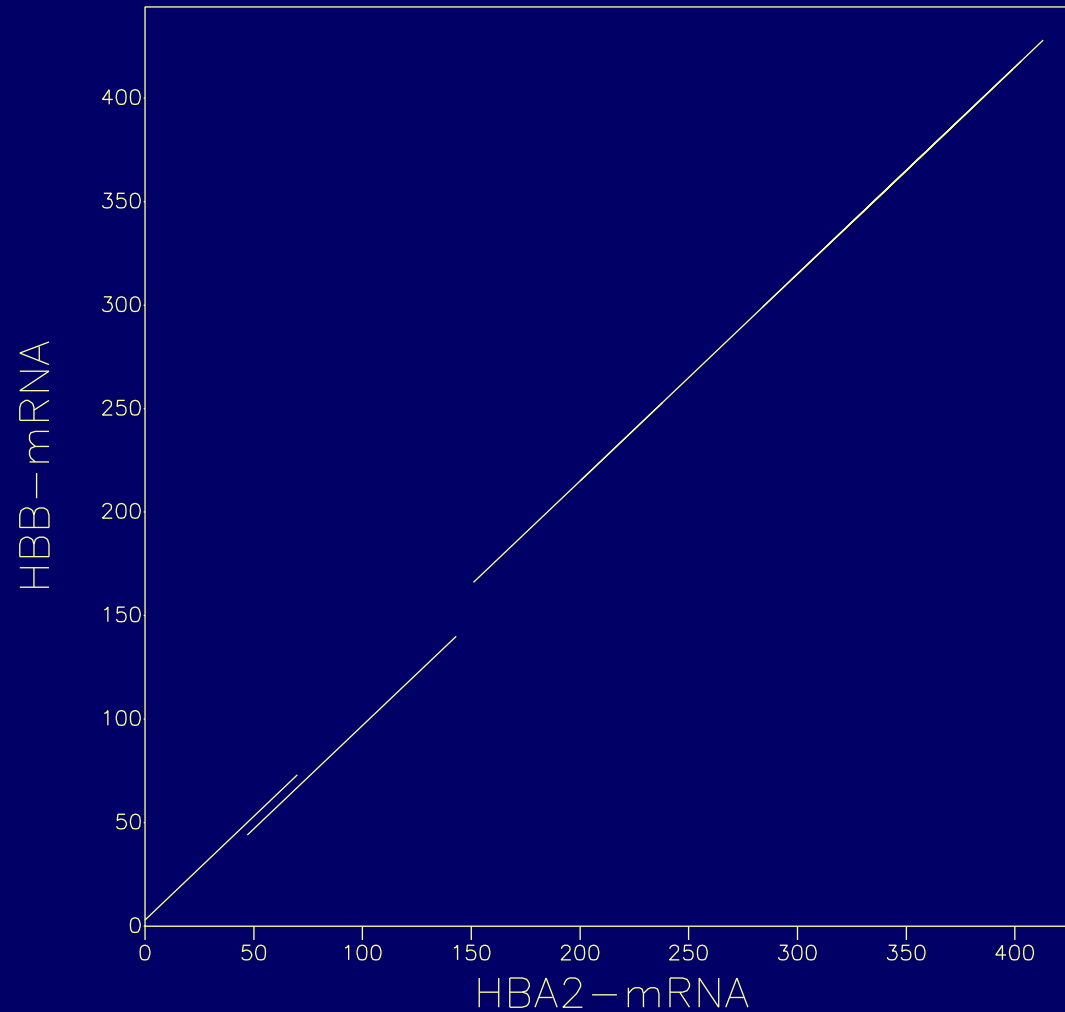
dotpath (08/06/05)



Taille fenêtre = 20

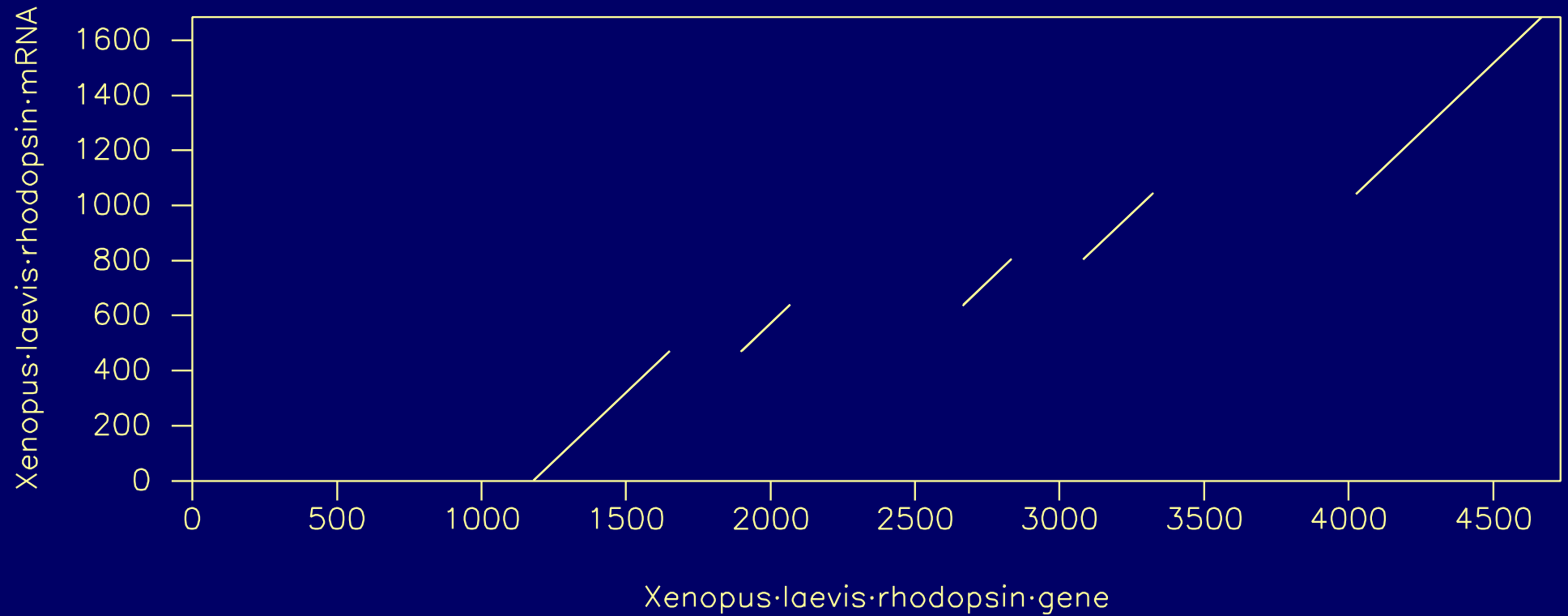
Dotmatcher: HBA2-mRNA vs HBB-mRNA

(windowsize = 60, threshold = 70.00 09/06/05)

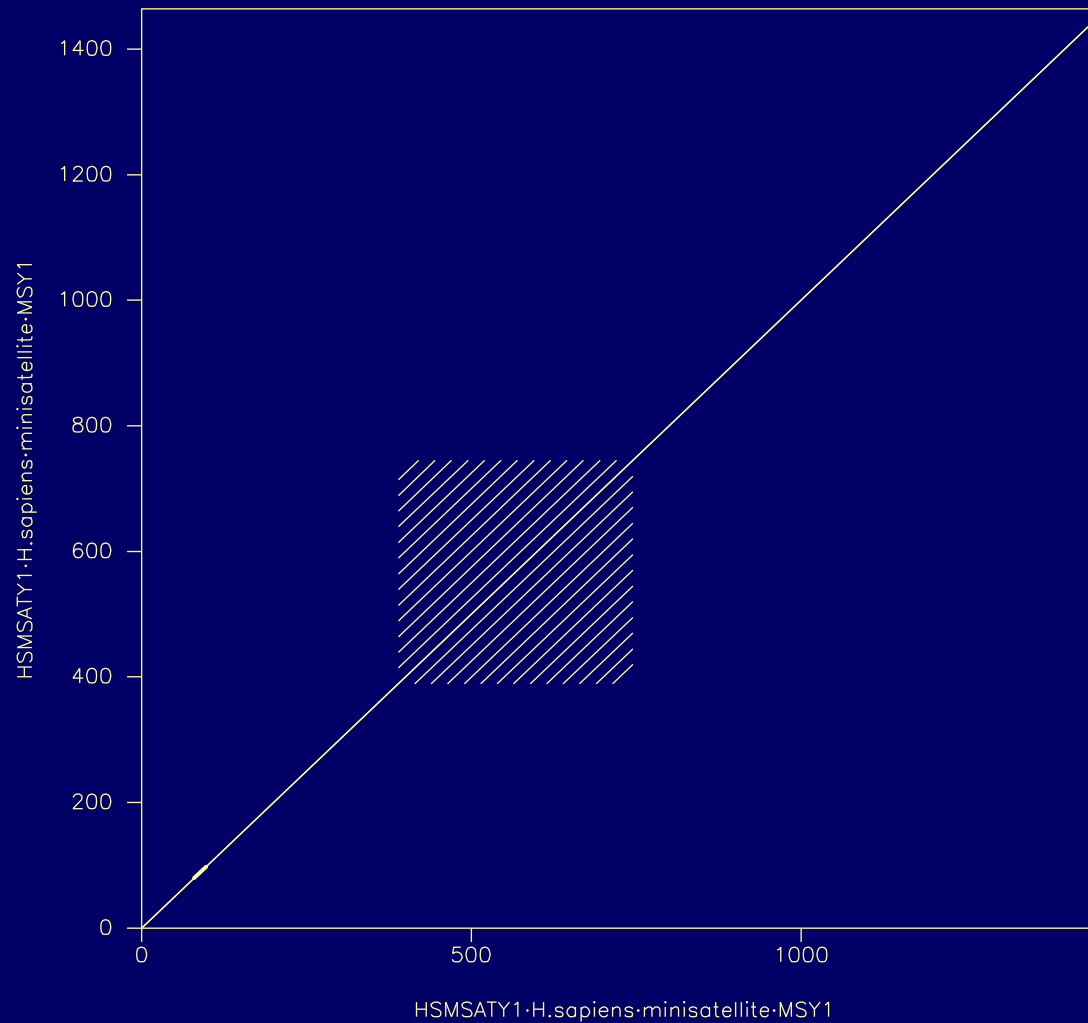


Hémoglobine humaine chaînes $\alpha 2$ et β

dotpath (08/06/05)



dotpath (09/06/05)



Minisatellite humain MSY1, taille de la fenêtre = 20

Avantages :

- simple, visuel
- très informatif

Inconvénients :

- identification \Rightarrow pas de méthode de détection automatique
- interprétation \Rightarrow pas de mesure objective

\Rightarrow Besoin d'une mesure quantitative de similarité

-
- Introduction
 - Dotplot
 - **Alignement : définitions et scores**
 - Algorithme d'alignement (programmation dynamique)
 - Autres types d'alignements
 - La suite ...

- Mise en correspondance de 2 séquences (ADN ou protéine)

<i>R</i>	<i>D</i>	<i>I</i>	<i>S</i>	<i>L</i>	<i>V</i>	-	-	-	<i>K</i>	<i>N</i>	<i>A</i>	<i>G</i>	<i>I</i>
<i>R</i>	<i>N</i>	<i>I</i>	-	<i>L</i>	<i>V</i>	<i>S</i>	<i>D</i>	<i>A</i>	<i>K</i>	<i>N</i>	<i>V</i>	<i>G</i>	<i>I</i>

- Mise en correspondance de 2 séquences (ADN ou protéine)

<i>R</i>	<i>D</i>	<i>I</i>	<i>S</i>	<i>L</i>	<i>V</i>	-	-	-	<i>K</i>	<i>N</i>	<i>A</i>	<i>G</i>	<i>I</i>
<i>R</i>	<i>N</i>	<i>I</i>	-	<i>L</i>	<i>V</i>	<i>S</i>	<i>D</i>	<i>A</i>	<i>K</i>	<i>N</i>	<i>V</i>	<i>G</i>	<i>I</i>

- 3 événements mutationnels élémentaires = 3 opérations

▷ substitution

▷ insertion } *indels*

▷ délétion }

- Mise en correspondance de 2 séquences (ADN ou protéine)

<i>R</i>	<i>D</i>	<i>I</i>	<i>S</i>	<i>L</i>	<i>V</i>	-	-	-	<i>K</i>	<i>N</i>	<i>A</i>	<i>G</i>	<i>I</i>
<i>R</i>	<i>N</i>	<i>I</i>	-	<i>L</i>	<i>V</i>	<i>S</i>	<i>D</i>	<i>A</i>	<i>K</i>	<i>N</i>	<i>V</i>	<i>G</i>	<i>I</i>

- 3 événements mutationnels élémentaires = 3 opérations

- ▷ substitution

- ▷ insertion
- ▷ délétion

} *indels*

- Donnée : une paire de séquences + une méthode de score/dist.
- But : Quantifier et localiser la similarité : score/dist. + alignement

- Mise en correspondance de 2 séquences (ADN ou protéine)

<i>R</i>	<i>D</i>	<i>I</i>	<i>S</i>	<i>L</i>	<i>V</i>	-	-	-	<i>K</i>	<i>N</i>	<i>A</i>	<i>G</i>	<i>I</i>
<i>R</i>	<i>N</i>	<i>I</i>	-	<i>L</i>	<i>V</i>	<i>S</i>	<i>D</i>	<i>A</i>	<i>K</i>	<i>N</i>	<i>V</i>	<i>G</i>	<i>I</i>

- 3 événements mutationnels élémentaires = 3 opérations

▷ substitution

▷ insertion } *indels*

▷ délétion }

- Donnée : une paire de séquences + une méthode de score/dist.
- But : Quantifier et localiser la similarité : score/dist. + alignement

⇒ Trouver la meilleure mise en correspondance = meilleur score/dist.

Soient s et t deux séquences respectivement de longueur n et m .

On note s_i le symbole à la position i de s .

Soient s et t deux séquences respectivement de longueur n et m .

On note s_i le symbole à la position i de s .

Définition : un **Alignement** global de s et t :

Soient s et t deux séquences respectivement de longueur n et m .
On note s_i le symbole à la position i de s .

Définition : un **Alignement** global de s et t :

→ une **matrice** à 2 lignes, et entre $\max(m, n)$ et $n + m$ colonnes

Soient s et t deux séquences respectivement de longueur n et m .
On note s_i le symbole à la position i de s .

Définition : un **Alignement** global de s et t :

→ une **matrice** à 2 lignes, et entre $\max(m, n)$ et $n + m$ colonnes

→ où chaque colonne met en correspondance deux symboles :

soit $\begin{pmatrix} s_i \\ t_j \end{pmatrix}$ ou $\begin{pmatrix} s_i \\ - \end{pmatrix}$ ou $\begin{pmatrix} - \\ t_j \end{pmatrix}$

Soient s et t deux séquences respectivement de longueur n et m .
On note s_i le symbole à la position i de s .

Définition : un **Alignement** global de s et t :

→ une **matrice** à 2 lignes, et entre $\max(m, n)$ et $n + m$ colonnes

→ où chaque colonne met en correspondance deux symboles :

soit $\begin{pmatrix} s_i \\ t_j \end{pmatrix}$ ou $\begin{pmatrix} s_i \\ - \end{pmatrix}$ ou $\begin{pmatrix} - \\ t_j \end{pmatrix}$

→ où les colonnes doivent respecter l'ordre des séquences

après $\begin{pmatrix} s_i \\ t_j \end{pmatrix}$ on trouve $\begin{pmatrix} s_{i+1} \\ - \end{pmatrix}$, $\begin{pmatrix} s_{i+1} \\ t_{j+1} \end{pmatrix}$ ou $\begin{pmatrix} - \\ t_{j+1} \end{pmatrix}$

Soient s et t deux séquences respectivement de longueur n et m .
On note s_i le symbole à la position i de s .

Définition : un **Alignement** global de s et t :

→ une **matrice** à 2 lignes, et entre $\max(m, n)$ et $n + m$ colonnes

→ où chaque colonne met en correspondance deux symboles :

$$\text{soit } \begin{pmatrix} s_i \\ t_j \end{pmatrix} \text{ ou } \begin{pmatrix} s_i \\ - \end{pmatrix} \text{ ou } \begin{pmatrix} - \\ t_j \end{pmatrix}$$

→ où les colonnes doivent respecter l'ordre des séquences

$$\text{après } \begin{pmatrix} s_i \\ t_j \end{pmatrix} \text{ on trouve } \begin{pmatrix} s_{i+1} \\ - \end{pmatrix}, \begin{pmatrix} s_{i+1} \\ t_{j+1} \end{pmatrix} \text{ ou } \begin{pmatrix} - \\ t_{j+1} \end{pmatrix}$$

Remarque : suivant le nombre de symboles $-$, le nombre de colonnes d'un alignement varie.

Séquences $s := I$, $t := L$

Alignement 1	Alignement 2	Alignement 3
$\begin{pmatrix} I & - \\ - & L \end{pmatrix}$	$\begin{pmatrix} - & I \\ L & - \end{pmatrix}$	$\begin{pmatrix} I \\ L \end{pmatrix}$

Séquences $s := I$, $t := L$

Alignement 1	Alignement 2	Alignement 3
$\begin{pmatrix} I & - \\ - & L \end{pmatrix}$	$\begin{pmatrix} - & I \\ L & - \end{pmatrix}$	$\begin{pmatrix} I \\ L \end{pmatrix}$

Séquences $s := IL$ et $t := LV \Rightarrow 13$ alignements possibles

Séquences $s := I, t := L$

Alignement 1	Alignement 2	Alignement 3
$\begin{pmatrix} I & - \\ - & L \end{pmatrix}$	$\begin{pmatrix} - & I \\ L & - \end{pmatrix}$	$\begin{pmatrix} I \\ L \end{pmatrix}$

Séquences $s := IL$ et $t := LV \Rightarrow 13$ alignements possibles

$\begin{pmatrix} I & L \\ L & V \end{pmatrix}$	$\begin{pmatrix} I & L & - \\ L & - & V \end{pmatrix}$	$\begin{pmatrix} I & L & - & - \\ - & - & L & V \end{pmatrix}$	$\begin{pmatrix} - & - & I & L \\ L & V & - & - \end{pmatrix}$
$\begin{pmatrix} I & - & L \\ - & L & V \end{pmatrix}$	$\begin{pmatrix} I & - & L \\ L & V & - \end{pmatrix}$	$\begin{pmatrix} I & - & L & - \\ - & L & - & V \end{pmatrix}$	$\begin{pmatrix} - & I & - & L \\ L & - & V & - \end{pmatrix}$
$\begin{pmatrix} I & L & - \\ - & L & V \end{pmatrix}$	$\begin{pmatrix} - & I & L \\ L & - & V \end{pmatrix}$	$\begin{pmatrix} I & - & - & L \\ - & L & V & - \end{pmatrix}$	$\begin{pmatrix} - & I & L & - \\ L & - & - & V \end{pmatrix}$
	$\begin{pmatrix} - & I & L \\ L & V & - \end{pmatrix}$		

- Identité/substitution : Scores positifs/négatifs
 - matrice s de score de similarité (PAM, BLOSUM, etc.)
 - $s(a, b)$ = score d'alignement des résidus a et b

- Identité/substitution : Scores positifs/négatifs
 - matrice s de score de similarité (PAM, BLOSUM, etc.)
 - $s(a, b) =$ score d'alignement des résidus a et b
- Insertion/délétion (indels) : Scores négatifs
 - fonction de pénalité $\left\{ \begin{array}{l} \text{élémentaire : unitaire pour un indel} \\ \text{complexe : affine, logarithmique} \end{array} \right.$

- Identité/substitution : Scores positifs/négatifs
 - matrice s de score de similarité (PAM, BLOSUM, etc.)
 - $s(a, b)$ = score d'alignement des résidus a et b
- Insertion/délétion (indels) : Scores négatifs
 - fonction de pénalité $\left\{ \begin{array}{l} \text{élémentaire : unitaire pour un indel} \\ \text{complexe : affine, logarithmique} \end{array} \right.$
- Score de l'alignement : somme des scores des événements élémentaires qui le composent

- Identité/substitution : Scores positifs/négatifs
 - matrice s de score de similarité (PAM, BLOSUM, etc.)
 - $s(a, b)$ = score d'alignement des résidus a et b
- Insertion/délétion (indels) : Scores négatifs
 - fonction de pénalité $\left\{ \begin{array}{l} \text{élémentaire : unitaire pour un indel} \\ \text{complexe : affine, logarithmique} \end{array} \right.$
- Score de l'alignement : somme des scores des événements élémentaires qui le composent

⇒ Maximiser le score

Distance d'édition

$$\text{Lev} \begin{pmatrix} a \\ b \end{pmatrix} = 1$$

$$\text{Lev} \begin{pmatrix} a \\ - \end{pmatrix} = 1$$

$$\text{Lev} \begin{pmatrix} - \\ b \end{pmatrix} = 1$$

Distance d'édition généralisée

$$\text{Edit} \begin{pmatrix} a \\ b \end{pmatrix} = \text{Sub}(a, b) \geq 0$$

$$\text{Edit} \begin{pmatrix} a \\ - \end{pmatrix} = \text{Del}(a) \geq 0$$

$$\text{Edit} \begin{pmatrix} - \\ b \end{pmatrix} = \text{Ins}(b) \geq 0$$

Seule l'identité = 0

Distance d'édition

$$Lev \begin{pmatrix} a \\ b \end{pmatrix} = 1$$

$$Lev \begin{pmatrix} a \\ - \end{pmatrix} = 1$$

$$Lev \begin{pmatrix} - \\ b \end{pmatrix} = 1$$

Distance d'édition généralisée

$$Edit \begin{pmatrix} a \\ b \end{pmatrix} = Sub(a, b) \geq 0$$

$$Edit \begin{pmatrix} a \\ - \end{pmatrix} = Del(a) \geq 0$$

$$Edit \begin{pmatrix} - \\ b \end{pmatrix} = Ins(b) \geq 0$$

Seule l'identité = 0

- **Coût de l'alignement** : somme des coûts des événements élémentaires qui le composent

Distance d'édition

$$\text{Lev} \begin{pmatrix} a \\ b \end{pmatrix} = 1$$

$$\text{Lev} \begin{pmatrix} a \\ - \end{pmatrix} = 1$$

$$\text{Lev} \begin{pmatrix} - \\ b \end{pmatrix} = 1$$

Distance d'édition généralisée

$$\text{Edit} \begin{pmatrix} a \\ b \end{pmatrix} = \text{Sub}(a, b) \geq 0$$

$$\text{Edit} \begin{pmatrix} a \\ - \end{pmatrix} = \text{Del}(a) \geq 0$$

$$\text{Edit} \begin{pmatrix} - \\ b \end{pmatrix} = \text{Ins}(b) \geq 0$$

Seule l'identité = 0

- **Coût de l'alignement** : somme des coûts des événements élémentaires qui le composent

⇒ Minimiser la distance

-
- Introduction
 - Dotplot
 - Alignement : définitions et scores
 - Algorithme d'alignement (programmation dynamique)
 - Autres types d'alignements
 - La suite ...

-
- Aligner les séquences : *ACGCTATC* et *ACTGTAATG*
 - Scores : indel = -1 , mismatch -2 , match = 2

-
- Aligner les séquences : *ACGCTATC* et *ACTGTAATG*
 - Scores : indel = -1 , mismatch -2 , match = 2
 - Alignement optimal complet, dernière opération : 3 possibilités

- Aligner les séquences : *ACGGCTATC* et *ACTGTAATG*
- Scores : indel = -1 , mismatch -2 , match = 2
- Alignement optimal complet, dernière opération : 3 possibilités

substitution

ACGGCTATC	C	-2
? ? ?		
ACTGTAATG	G	

délétion

ACGGCTAT	C	-1
? ? ?		
ACTGTAATG	-	

insertion

ACGGCTATC	-	-1
? ? ?		
ACTGTAAT	G	

- Aligner les séquences : *ACGGCTATC* et *ACTGTAATG*
- Scores : indel = -1 , mismatch -2 , match = 2
- Alignement optimal complet, dernière opération : 3 possibilités

substitution

ACGGCTATC	C	-2
? ? ?		
ACTGTAATG	G	

délétion

ACGGCTAT	C	-1
? ? ?		
ACTGTAATG	-	

insertion

ACGGCTATC	-	-1
? ? ?		
ACTGTAAT	G	

Problème sur des séquences plus courtes

- Aligner les séquences : *ACGGCTATC* et *ACTGTAATG*
- Scores : indel = -1 , mismatch -2 , match = 2
- Alignement optimal complet, dernière opération : 3 possibilités

substitution

ACGGCTATC	C	-2
? ? ?		
ACTGTAATG	G	

délétion

ACGGCTAT	C	-1
? ? ?		
ACTGTAATG	-	

insertion

ACGGCTATC	-	-1
? ? ?		
ACTGTAAT	G	

Problème sur des séquences plus courtes

⇒ Récurrence, programmation dynamique

-
- La **programmation dynamique** résout les problèmes en combinant les solutions de **sous-problèmes**
 - résout chaque sous-problème **1 seule fois**
 - mémorise sa solution dans une **matrice** (de prog. dyn.)
(épargnant ainsi le recalcul de la sol. chaque fois que le sous-pb est rencontré)

- La **programmation dynamique** résout les problèmes en combinant les solutions de **sous-problèmes**
 - résout chaque sous-problème **1 seule fois**
 - mémorise sa solution dans une **matrice** (de prog. dyn.)
(épargnant ainsi le recalcul de la sol. chaque fois que le sous-pb est rencontré)

Ici :

calculs intermédiaires

=

scores d'alignements entre préfixes

		0	1	2	3	4	5	6	7	8	9
			<i>A</i>	<i>C</i>	<i>G</i>	<i>G</i>	<i>C</i>	<i>T</i>	<i>A</i>	<i>T</i>	<i>C</i>
0											
1	<i>A</i>										
2	<i>C</i>										
3	<i>T</i>										
4	<i>G</i>										
5	<i>T</i>										
6	<i>A</i>										
7	<i>A</i>										
8	<i>T</i>										
9	<i>G</i>										

Création d'une table indexée par les deux séquences

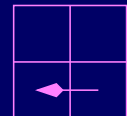
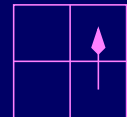
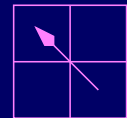
Case (i, j) : score alignement entre les i premières bases de *ACTGTAATG* et les j premières bases de *ACGGCTATC*

- s : une matrice de score ; g : pénalité associée à un indel

- Initialisation : $M(0,0) = 0$, $M(0,j) = g \times j$, $M(i,0) = g \times i$

- Remplissage

$$M(i, j) = \max \begin{cases} M(i-1, j-1) + s(x_i, y_j) & \text{match/mismatch} \\ M(i-1, j) + g & \text{délétion} \\ M(i, j-1) + g & \text{insertion} \end{cases}$$



		<i>A</i>	<i>C</i>	<i>G</i>	<i>G</i>	<i>C</i>	<i>T</i>	<i>A</i>	<i>T</i>	<i>C</i>
<i>A</i>										
<i>C</i>										
<i>T</i>										
<i>G</i>										
<i>T</i>										
<i>A</i>										
<i>A</i>										
<i>T</i>										
<i>G</i>										

Coûts : $s(a,b) = -1$ avec $a \neq b$ et 2 sinon ; $g = -1$

		<i>A</i>	<i>C</i>	<i>G</i>	<i>G</i>	<i>C</i>	<i>T</i>	<i>A</i>	<i>T</i>	<i>C</i>
	0									
<i>A</i>										
<i>C</i>										
<i>T</i>										
<i>G</i>										
<i>T</i>										
<i>A</i>										
<i>A</i>										
<i>T</i>										
<i>G</i>										

Coûts : $s(a,b) = -1$ avec $a \neq b$ et 2 sinon ; $g = -1$

		<i>A</i>	<i>C</i>	<i>G</i>	<i>G</i>	<i>C</i>	<i>T</i>	<i>A</i>	<i>T</i>	<i>C</i>
	0	-1								
<i>A</i>										
<i>C</i>										
<i>T</i>										
<i>G</i>										
<i>T</i>										
<i>A</i>										
<i>A</i>										
<i>T</i>										
<i>G</i>										

Coûts : $s(a,b) = -1$ avec $a \neq b$ et 2 sinon ; $g = -1$

		<i>A</i>	<i>C</i>	<i>G</i>	<i>G</i>	<i>C</i>	<i>T</i>	<i>A</i>	<i>T</i>	<i>C</i>
	0	-1	-2							
<i>A</i>										
<i>C</i>										
<i>T</i>										
<i>G</i>										
<i>T</i>										
<i>A</i>										
<i>A</i>										
<i>T</i>										
<i>G</i>										

Coûts : $s(a,b) = -1$ avec $a \neq b$ et 2 sinon ; $g = -1$

		<i>A</i>	<i>C</i>	<i>G</i>	<i>G</i>	<i>C</i>	<i>T</i>	<i>A</i>	<i>T</i>	<i>C</i>
	0	-1	-2	-3						
<i>A</i>										
<i>C</i>										
<i>T</i>										
<i>G</i>										
<i>T</i>										
<i>A</i>										
<i>A</i>										
<i>T</i>										
<i>G</i>										

Coûts : $s(a,b) = -1$ avec $a \neq b$ et 2 sinon ; $g = -1$

		<i>A</i>	<i>C</i>	<i>G</i>	<i>G</i>	<i>C</i>	<i>T</i>	<i>A</i>	<i>T</i>	<i>C</i>
	0	-1	-2	-3	-4	-5	-6	-7	-8	-9
<i>A</i>										
<i>C</i>										
<i>T</i>										
<i>G</i>										
<i>T</i>										
<i>A</i>										
<i>A</i>										
<i>T</i>										
<i>G</i>										

Coûts : $s(a,b) = -1$ avec $a \neq b$ et 2 sinon ; $g = -1$

		<i>A</i>	<i>C</i>	<i>G</i>	<i>G</i>	<i>C</i>	<i>T</i>	<i>A</i>	<i>T</i>	<i>C</i>
	0	-1	-2	-3	-4	-5	-6	-7	-8	-9
<i>A</i>	-1									
<i>C</i>										
<i>T</i>										
<i>G</i>										
<i>T</i>										
<i>A</i>										
<i>A</i>										
<i>T</i>										
<i>G</i>										

Coûts : $s(a,b) = -1$ avec $a \neq b$ et 2 sinon ; $g = -1$

		<i>A</i>	<i>C</i>	<i>G</i>	<i>G</i>	<i>C</i>	<i>T</i>	<i>A</i>	<i>T</i>	<i>C</i>
	0	-1	-2	-3	-4	-5	-6	-7	-8	-9
<i>A</i>	-1									
<i>C</i>	-2									
<i>T</i>										
<i>G</i>										
<i>T</i>										
<i>A</i>										
<i>A</i>										
<i>T</i>										
<i>G</i>										

Coûts : $s(a,b) = -1$ avec $a \neq b$ et 2 sinon ; $g = -1$

		<i>A</i>	<i>C</i>	<i>G</i>	<i>G</i>	<i>C</i>	<i>T</i>	<i>A</i>	<i>T</i>	<i>C</i>
	0	-1	-2	-3	-4	-5	-6	-7	-8	-9
<i>A</i>	-1									
<i>C</i>	-2									
<i>T</i>	-3									
<i>G</i>	-4									
<i>T</i>	-5									
<i>A</i>	-6									
<i>A</i>	-7									
<i>T</i>	-8									
<i>G</i>	-9									

Coûts : $s(a,b) = -1$ avec $a \neq b$ et 2 sinon ; $g = -1$

		<i>A</i>	<i>C</i>	<i>G</i>	<i>G</i>	<i>C</i>	<i>T</i>	<i>A</i>	<i>T</i>	<i>C</i>
	0	-1	-2	-3	-4	-5	-6	-7	-8	-9
<i>A</i>	-1	2								
<i>C</i>	-2									
<i>T</i>	-3									
<i>G</i>	-4									
<i>T</i>	-5									
<i>A</i>	-6									
<i>A</i>	-7									
<i>T</i>	-8									
<i>G</i>	-9									

Coûts : $s(a,b) = -1$ avec $a \neq b$ et 2 sinon ; $g = -1$

		<i>A</i>	<i>C</i>	<i>G</i>	<i>G</i>	<i>C</i>	<i>T</i>	<i>A</i>	<i>T</i>	<i>C</i>
	0	-1	-2	-3	-4	-5	-6	-7	-8	-9
<i>A</i>	-1	2	1							
<i>C</i>	-2									
<i>T</i>	-3									
<i>G</i>	-4									
<i>T</i>	-5									
<i>A</i>	-6									
<i>A</i>	-7									
<i>T</i>	-8									
<i>G</i>	-9									

Coûts : $s(a,b) = -1$ avec $a \neq b$ et 2 sinon ; $g = -1$

		<i>A</i>	<i>C</i>	<i>G</i>	<i>G</i>	<i>C</i>	<i>T</i>	<i>A</i>	<i>T</i>	<i>C</i>
	0	-1	-2	-3	-4	-5	-6	-7	-8	-9
<i>A</i>	-1	2	1	0						
<i>C</i>	-2									
<i>T</i>	-3									
<i>G</i>	-4									
<i>T</i>	-5									
<i>A</i>	-6									
<i>A</i>	-7									
<i>T</i>	-8									
<i>G</i>	-9									

Coûts : $s(a,b) = -1$ avec $a \neq b$ et 2 sinon ; $g = -1$

		<i>A</i>	<i>C</i>	<i>G</i>	<i>G</i>	<i>C</i>	<i>T</i>	<i>A</i>	<i>T</i>	<i>C</i>
	0	-1	-2	-3	-4	-5	-6	-7	-8	-9
<i>A</i>	-1	2	1	0	-1					
<i>C</i>	-2									
<i>T</i>	-3									
<i>G</i>	-4									
<i>T</i>	-5									
<i>A</i>	-6									
<i>A</i>	-7									
<i>T</i>	-8									
<i>G</i>	-9									

Coûts : $s(a,b) = -1$ avec $a \neq b$ et 2 sinon ; $g = -1$

		<i>A</i>	<i>C</i>	<i>G</i>	<i>G</i>	<i>C</i>	<i>T</i>	<i>A</i>	<i>T</i>	<i>C</i>
	0	-1	-2	-3	-4	-5	-6	-7	-8	-9
<i>A</i>	-1	2	1	0	-1	-2	-3	-4	-5	-6
<i>C</i>	-2									
<i>T</i>	-3									
<i>G</i>	-4									
<i>T</i>	-5									
<i>A</i>	-6									
<i>A</i>	-7									
<i>T</i>	-8									
<i>G</i>	-9									

Coûts : $s(a,b) = -1$ avec $a \neq b$ et 2 sinon ; $g = -1$

		<i>A</i>	<i>C</i>	<i>G</i>	<i>G</i>	<i>C</i>	<i>T</i>	<i>A</i>	<i>T</i>	<i>C</i>
	0	-1	-2	-3	-4	-5	-6	-7	-8	-9
<i>A</i>	-1	2	1	0	-1	-2	-3	-4	-5	-6
<i>C</i>	-2	1								
<i>T</i>	-3									
<i>G</i>	-4									
<i>T</i>	-5									
<i>A</i>	-6									
<i>A</i>	-7									
<i>T</i>	-8									
<i>G</i>	-9									

Coûts : $s(a,b) = -1$ avec $a \neq b$ et 2 sinon ; $g = -1$

		<i>A</i>	<i>C</i>	<i>G</i>	<i>G</i>	<i>C</i>	<i>T</i>	<i>A</i>	<i>T</i>	<i>C</i>
	0	-1	-2	-3	-4	-5	-6	-7	-8	-9
<i>A</i>	-1	2	1	0	-1	-2	-3	-4	-5	-6
<i>C</i>	-2	1	4							
<i>T</i>	-3									
<i>G</i>	-4									
<i>T</i>	-5									
<i>A</i>	-6									
<i>A</i>	-7									
<i>T</i>	-8									
<i>G</i>	-9									

Coûts : $s(a,b) = -1$ avec $a \neq b$ et 2 sinon ; $g = -1$

		<i>A</i>	<i>C</i>	<i>G</i>	<i>G</i>	<i>C</i>	<i>T</i>	<i>A</i>	<i>T</i>	<i>C</i>
	0	-1	-2	-3	-4	-5	-6	-7	-8	-9
<i>A</i>	-1	2	1	0	-1	-2	-3	-4	-5	-6
<i>C</i>	-2	1	4	3						
<i>T</i>	-3									
<i>G</i>	-4									
<i>T</i>	-5									
<i>A</i>	-6									
<i>A</i>	-7									
<i>T</i>	-8									
<i>G</i>	-9									

Coûts : $s(a,b) = -1$ avec $a \neq b$ et 2 sinon ; $g = -1$

		<i>A</i>	<i>C</i>	<i>G</i>	<i>G</i>	<i>C</i>	<i>T</i>	<i>A</i>	<i>T</i>	<i>C</i>
	0	-1	-2	-3	-4	-5	-6	-7	-8	-9
<i>A</i>	-1	2	1	0	-1	-2	-3	-4	-5	-6
<i>C</i>	-2	1	4	3	2	1	0	-1	-2	-3
<i>T</i>	-3									
<i>G</i>	-4									
<i>T</i>	-5									
<i>A</i>	-6									
<i>A</i>	-7									
<i>T</i>	-8									
<i>G</i>	-9									

Coûts : $s(a,b) = -1$ avec $a \neq b$ et 2 sinon ; $g = -1$

		<i>A</i>	<i>C</i>	<i>G</i>	<i>G</i>	<i>C</i>	<i>T</i>	<i>A</i>	<i>T</i>	<i>C</i>
	0	-1	-2	-3	-4	-5	-6	-7	-8	-9
<i>A</i>	-1	2	1	0	-1	-2	-3	-4	-5	-6
<i>C</i>	-2	1	4	3	2	1	0	-1	-2	-3
<i>T</i>	-3	0								
<i>G</i>	-4									
<i>T</i>	-5									
<i>A</i>	-6									
<i>A</i>	-7									
<i>T</i>	-8									
<i>G</i>	-9									

Coûts : $s(a,b) = -1$ avec $a \neq b$ et 2 sinon ; $g = -1$

		<i>A</i>	<i>C</i>	<i>G</i>	<i>G</i>	<i>C</i>	<i>T</i>	<i>A</i>	<i>T</i>	<i>C</i>
	0	-1	-2	-3	-4	-5	-6	-7	-8	-9
<i>A</i>	-1	2	1	0	-1	-2	-3	-4	-5	-6
<i>C</i>	-2	1	4	3	2	1	0	-1	-2	-3
<i>T</i>	-3	0	3							
<i>G</i>	-4									
<i>T</i>	-5									
<i>A</i>	-6									
<i>A</i>	-7									
<i>T</i>	-8									
<i>G</i>	-9									

Coûts : $s(a,b) = -1$ avec $a \neq b$ et 2 sinon ; $g = -1$

		<i>A</i>	<i>C</i>	<i>G</i>	<i>G</i>	<i>C</i>	<i>T</i>	<i>A</i>	<i>T</i>	<i>C</i>
	0	-1	-2	-3	-4	-5	-6	-7	-8	-9
<i>A</i>	-1	2	1	0	-1	-2	-3	-4	-5	-6
<i>C</i>	-2	1	4	3	2	1	0	-1	-2	-3
<i>T</i>	-3	0	3	3	2	1	3	2	1	0
<i>G</i>	-4									
<i>T</i>	-5									
<i>A</i>	-6									
<i>A</i>	-7									
<i>T</i>	-8									
<i>G</i>	-9									

Coûts : $s(a,b) = -1$ avec $a \neq b$ et 2 sinon ; $g = -1$

		<i>A</i>	<i>C</i>	<i>G</i>	<i>G</i>	<i>C</i>	<i>T</i>	<i>A</i>	<i>T</i>	<i>C</i>
	0	-1	-2	-3	-4	-5	-6	-7	-8	-9
<i>A</i>	-1	2	1	0	-1	-2	-3	-4	-5	-6
<i>C</i>	-2	1	4	3	2	1	0	-1	-2	-3
<i>T</i>	-3	0	3	3	2	1	3	2	1	0
<i>G</i>	-4	-1	2	5	5	4	3	2	1	0
<i>T</i>	-5	-2	1	4	4	4	6	5	4	3
<i>A</i>	-6	-3	0	3	3	3	5	8	7	6
<i>A</i>	-7	-4	-1	2	2	2	4	7	7	6
<i>T</i>	-8	-5	-2	1	1	1	4	6	9	8
<i>G</i>	-9	-6	-3	0	3	2	3	5	8	8

Coûts : $s(a, b) = -1$ avec $a \neq b$ et 2 sinon ; $g = -1$

		A	C	G	G	C	T	A	T	C
	0	-1	-2	-3	-4	-5	-6	-7	-8	-9
A	-1	2	1	0	-1	-2	-3	-4	-5	-6
C	-2	1	4	3	2	1	0	-1	-2	-3
T	-3	0	3	3	2	1	3	2	1	0
G	-4	-1	2	5	5	4	3	2	1	0
T	-5	-2	1	4	4	4	6	5	4	3
A	-6	-3	0	3	3	3	5	8	7	6
A	-7	-4	-1	2	2	2	4	7	7	6
T	-8	-5	-2	1	1	1	4	6	9	8
G	-9	-6	-3	0	3	2	3	5	8	8

Coûts : $s(a,b) = -1$ avec $a \neq b$ et 2 sinon ; $g = -1$

Case (9,9) : score de l'alignement global entre *ACGGCTATC* et *ACTGTAATG*.

-
- Procédure qui permet de trouver l'alignement en fonction de la matrice
 - Fonctionnement :
 1. À partir de la cellule d'arrivée, remonter vers la(les) cellule(s) voisine(s) de score maximal et telle que son score plus la mutation correspondante donne le score de la cellule courante
 2. Itérer jusqu'à arriver à la cellule initiale.
 - Si en une cellule, on peut revenir vers plusieurs cellules voisines, alors il existe plusieurs chemins optimaux.

		<i>A</i>	<i>C</i>	<i>G</i>	<i>G</i>	<i>C</i>	<i>T</i>	<i>A</i>	<i>T</i>	<i>C</i>
	0	-1	-2	-3	-4	-5	-6	-7	-8	-9
<i>A</i>	-1	2	1	0	-1	-2	-3	-4	-5	-6
<i>C</i>	-2	1	4	3	2	1	0	-1	-2	-3
<i>T</i>	-3	0	3	3	2	1	3	2	1	0
<i>G</i>	-4	-1	2	5	5	4	3	2	1	0
<i>T</i>	-5	-2	1	4	4	4	6	5	4	3
<i>A</i>	-6	-3	0	3	3	3	5	8	7	6
<i>A</i>	-7	-4	-1	2	2	2	4	7	7	6
<i>T</i>	-8	-5	-2	1	1	1	4	6	9	8
<i>G</i>	-9	-6	-3	0	3	2	3	5	8	8

		<i>A</i>	<i>C</i>	<i>G</i>	<i>G</i>	<i>C</i>	<i>T</i>	<i>A</i>	<i>T</i>	<i>C</i>
	0	-1	-2	-3	-4	-5	-6	-7	-8	-9
<i>A</i>	-1	2	1	0	-1	-2	-3	-4	-5	-6
<i>C</i>	-2	1	4	3	2	1	0	-1	-2	-3
<i>T</i>	-3	0	3	3	2	1	3	2	1	0
<i>G</i>	-4	-1	2	5	5	4	3	2	1	0
<i>T</i>	-5	-2	1	4	4	4	6	5	4	3
<i>A</i>	-6	-3	0	3	3	3	5	8	7	6
<i>A</i>	-7	-4	-1	2	2	2	4	7	7	6
<i>T</i>	-8	-5	-2	1	1	1	4	6	9	8
<i>G</i>	-9	-6	-3	0	3	2	3	5	8	8

G

C

		<i>A</i>	<i>C</i>	<i>G</i>	<i>G</i>	<i>C</i>	<i>T</i>	<i>A</i>	<i>T</i>	<i>C</i>
	0	-1	-2	-3	-4	-5	-6	-7	-8	-9
<i>A</i>	-1	2	1	0	-1	-2	-3	-4	-5	-6
<i>C</i>	-2	1	4	3	2	1	0	-1	-2	-3
<i>T</i>	-3	0	3	3	2	1	3	2	1	0
<i>G</i>	-4	-1	2	5	5	4	3	2	1	0
<i>T</i>	-5	-2	1	4	4	4	6	5	4	3
<i>A</i>	-6	-3	0	3	3	3	5	8	7	6
<i>A</i>	-7	-4	-1	2	2	2	4	7	7	6
<i>T</i>	-8	-5	-2	1	1	1	4	6	9	8
<i>G</i>	-9	-6	-3	0	3	2	3	5	8	8

T *G*
 |
T *C*

		<i>A</i>	<i>C</i>	<i>G</i>	<i>G</i>	<i>C</i>	<i>T</i>	<i>A</i>	<i>T</i>	<i>C</i>
	0	-1	-2	-3	-4	-5	-6	-7	-8	-9
<i>A</i>	-1	2	1	0	-1	-2	-3	-4	-5	-6
<i>C</i>	-2	1	4	3	2	1	0	-1	-2	-3
<i>T</i>	-3	0	3	3	2	1	3	2	1	0
<i>G</i>	-4	-1	2	5	5	4	3	2	1	0
<i>T</i>	-5	-2	1	4	4	4	6	5	4	3
<i>A</i>	-6	-3	0	3	3	3	5	8	7	6
<i>A</i>	-7	-4	-1	2	2	2	4	7	7	6
<i>T</i>	-8	-5	-2	1	1	1	4	6	9	8
<i>G</i>	-9	-6	-3	0	3	2	3	5	8	8

A T G
 | |
A T C

		<i>A</i>	<i>C</i>	<i>G</i>	<i>G</i>	<i>C</i>	<i>T</i>	<i>A</i>	<i>T</i>	<i>C</i>
	0	-1	-2	-3	-4	-5	-6	-7	-8	-9
<i>A</i>	-1	2	1	0	-1	-2	-3	-4	-5	-6
<i>C</i>	-2	1	4	3	2	1	0	-1	-2	-3
<i>T</i>	-3	0	3	3	2	1	3	2	1	0
<i>G</i>	-4	-1	2	5	5	4	3	2	1	0
<i>T</i>	-5	-2	1	4	4	4	6	5	4	3
<i>A</i>	-6	-3	0	3	3	3	5	8	7	6
<i>A</i>	-7	-4	-1	2	2	2	4	7	7	6
<i>T</i>	-8	-5	-2	1	1	1	4	6	9	8
<i>G</i>	-9	-6	-3	0	3	2	3	5	8	8

A A T G

| |

- *A T C*

		<i>A</i>	<i>C</i>	<i>G</i>	<i>G</i>	<i>C</i>	<i>T</i>	<i>A</i>	<i>T</i>	<i>C</i>
	0	-1	-2	-3	-4	-5	-6	-7	-8	-9
<i>A</i>	-1	2	1	0	-1	-2	-3	-4	-5	-6
<i>C</i>	-2	1	4	3	2	1	0	-1	-2	-3
<i>T</i>	-3	0	3	3	2	1	3	2	1	0
<i>G</i>	-4	-1	2	5	5	4	3	2	1	0
<i>T</i>	-5	-2	1	4	4	4	6	5	4	3
<i>A</i>	-6	-3	0	3	3	3	5	8	7	6
<i>A</i>	-7	-4	-1	2	2	2	4	7	7	6
<i>T</i>	-8	-5	-2	1	1	1	4	6	9	8
<i>G</i>	-9	-6	-3	0	3	2	3	5	8	8

T *A* *A* *T* *G*
 | | |
T - *A* *T* *C*

		<i>A</i>	<i>C</i>	<i>G</i>	<i>G</i>	<i>C</i>	<i>T</i>	<i>A</i>	<i>T</i>	<i>C</i>
	0	-1	-2	-3	-4	-5	-6	-7	-8	-9
<i>A</i>	-1	2	1	0	-1	-2	-3	-4	-5	-6
<i>C</i>	-2	1	4	3	2	1	0	-1	-2	-3
<i>T</i>	-3	0	3	3	2	1	3	2	1	0
<i>G</i>	-4	-1	2	5	5	4	3	2	1	0
<i>T</i>	-5	-2	1	4	4	4	6	5	4	3
<i>A</i>	-6	-3	0	3	3	3	5	8	7	6
<i>A</i>	-7	-4	-1	2	2	2	4	7	7	6
<i>T</i>	-8	-5	-2	1	1	1	4	6	9	8
<i>G</i>	-9	-6	-3	0	3	2	3	5	8	8

- *T* *A* *A* *T* *G*
 | | |
C *T* - *A* *T* *C*

		<i>A</i>	<i>C</i>	<i>G</i>	<i>G</i>	<i>C</i>	<i>T</i>	<i>A</i>	<i>T</i>	<i>C</i>
	0	-1	-2	-3	-4	-5	-6	-7	-8	-9
<i>A</i>	-1	2	1	0	-1	-2	-3	-4	-5	-6
<i>C</i>	-2	1	4	3	2	1	0	-1	-2	-3
<i>T</i>	-3	0	3	3	2	1	3	2	1	0
<i>G</i>	-4	-1	2	5	5	4	3	2	1	0
<i>T</i>	-5	-2	1	4	4	4	6	5	4	3
<i>A</i>	-6	-3	0	3	3	3	5	8	7	6
<i>A</i>	-7	-4	-1	2	2	2	4	7	7	6
<i>T</i>	-8	-5	-2	1	1	1	4	6	9	8
<i>G</i>	-9	-6	-3	0	3	2	3	5	8	8

G - *T* *A* *A* *T* *G*
 | | | |
G *C* *T* - *A* *T* *C*

		A	C	G	G	C	T	A	T	C
	0	-1	-2	-3	-4	-5	-6	-7	-8	-9
A	-1	2	1	0	-1	-2	-3	-4	-5	-6
C	-2	1	4	3	2	1	0	-1	-2	-3
T	-3	0	3	3	2	1	3	2	1	0
G	-4	-1	2	5	5	4	3	2	1	0
T	-5	-2	1	4	4	4	6	5	4	3
A	-6	-3	0	3	3	3	5	8	7	6
A	-7	-4	-1	2	2	2	4	7	7	6
T	-8	-5	-2	1	1	1	4	6	9	8
G	-9	-6	-3	0	3	2	3	5	8	8

T G - T A A T G
 | | | |
G G C T - A T C

		<i>A</i>	<i>C</i>	<i>G</i>	<i>G</i>	<i>C</i>	<i>T</i>	<i>A</i>	<i>T</i>	<i>C</i>
	0	-1	-2	-3	-4	-5	-6	-7	-8	-9
<i>A</i>	-1	2	1	0	-1	-2	-3	-4	-5	-6
<i>C</i>	-2	1	4	3	2	1	0	-1	-2	-3
<i>T</i>	-3	0	3	3	2	1	3	2	1	0
<i>G</i>	-4	-1	2	5	5	4	3	2	1	0
<i>T</i>	-5	-2	1	4	4	4	6	5	4	3
<i>A</i>	-6	-3	0	3	3	3	5	8	7	6
<i>A</i>	-7	-4	-1	2	2	2	4	7	7	6
<i>T</i>	-8	-5	-2	1	1	1	4	6	9	8
<i>G</i>	-9	-6	-3	0	3	2	3	5	8	8

C T G - T A A T G
 | | | | |
C G G C T - A T C

		<i>A</i>	<i>C</i>	<i>G</i>	<i>G</i>	<i>C</i>	<i>T</i>	<i>A</i>	<i>T</i>	<i>C</i>
	0	-1	-2	-3	-4	-5	-6	-7	-8	-9
<i>A</i>	-1	2	1	0	-1	-2	-3	-4	-5	-6
<i>C</i>	-2	1	4	3	2	1	0	-1	-2	-3
<i>T</i>	-3	0	3	3	2	1	3	2	1	0
<i>G</i>	-4	-1	2	5	5	4	3	2	1	0
<i>T</i>	-5	-2	1	4	4	4	6	5	4	3
<i>A</i>	-6	-3	0	3	3	3	5	8	7	6
<i>A</i>	-7	-4	-1	2	2	2	4	7	7	6
<i>T</i>	-8	-5	-2	1	1	1	4	6	9	8
<i>G</i>	-9	-6	-3	0	3	2	3	5	8	8

A C T G _ T A A T G
 | | | | | | | |
A C G G C T _ A T C

		<i>G</i>	<i>G</i>	<i>C</i>	<i>T</i>	<i>G</i>	<i>A</i>	<i>C</i>
<i>G</i>								
<i>A</i>								
<i>T</i>								
<i>C</i>								

Initialisation

$$M(0, 0) = 0$$

$$M(0, j) = g \times j$$

$$M(i, 0) = g \times i$$

Remplissage

$$M(i, j) =$$

$$\min \begin{cases} M(i-1, j-1) + s(x_i, y_j) & \text{match/mismatch} \\ M(i-1, j) + g & \text{délétion} \\ M(i, j-1) + g & \text{insertion} \end{cases}$$

ici $g = 3$ *et* $s(a, b) = 3$ *si* $a \neq b$ *et* 0 *si* $a = b$

Les matrices protéiques

- Propriétés physico-chimiques : polarité, masse, etc.
pas convaincant à l'usage
- Observation des fréquences de mutation lors de l'évolution
 - Matrices PAM [Dayoff, 1969] Fréquence de changement entre A.A., bien pour ancêtre commun, PAM N : N mut. acceptées par 100 A.A
 - Matrices BLOSUM [Henikoff&Henikoff, 1992] Fréquence de changement entre A.A. avec conservation de structure, bien pour sim. loc., BLOSUM N : seuil de similarité

- Séquences protéiques d'enzymes dépendantes de la thiamine phosphatase (TPP) : `ILVB_ARATH` et `DCP1_MAIZE`
- Alignement global avec des pénalités 12/2 et les matrices PAM30, PAM60 et PAM350.
- Logiciel : `stretcher` de la suite `EMBOSS`
- **Résultats** : Le pourcentage d'identité fluctue autour de 20%. Le score passe de -977 à -511, puis à 396. L'allure de l'alignement est aussi nettement amélioré.

- Nombre d'alignements possibles : **exponentiel**
- Mémoire et temps utilisés $\mathcal{O}(n \times m)$ c'est-à-dire proportionnels au produit de la longueur des séquences
- **Amélioration** : algorithme linéaire en mémoire, algorithme de k -band

-
- Introduction
 - Dotplot
 - Alignement : définitions et scores
 - Algorithme d'alignement (programmation dynamique)
 - **Autres types d'alignements**
 - La suite ...

- Principe : alignement des meilleures sous-séquences
[Smith-Waterman 81]

... g q v a r y a g	E	K	L	F	H	S	I	F	V	E	q n i f s l t ...
... t e x l i n y i	E	K	L	F	V	-	L	R	V	E	l a e s a s ...

- Évaluation d'une **ressemblance locale** entre deux séquences ;
- Recherche de la région de plus forte similarité.

Alignement global

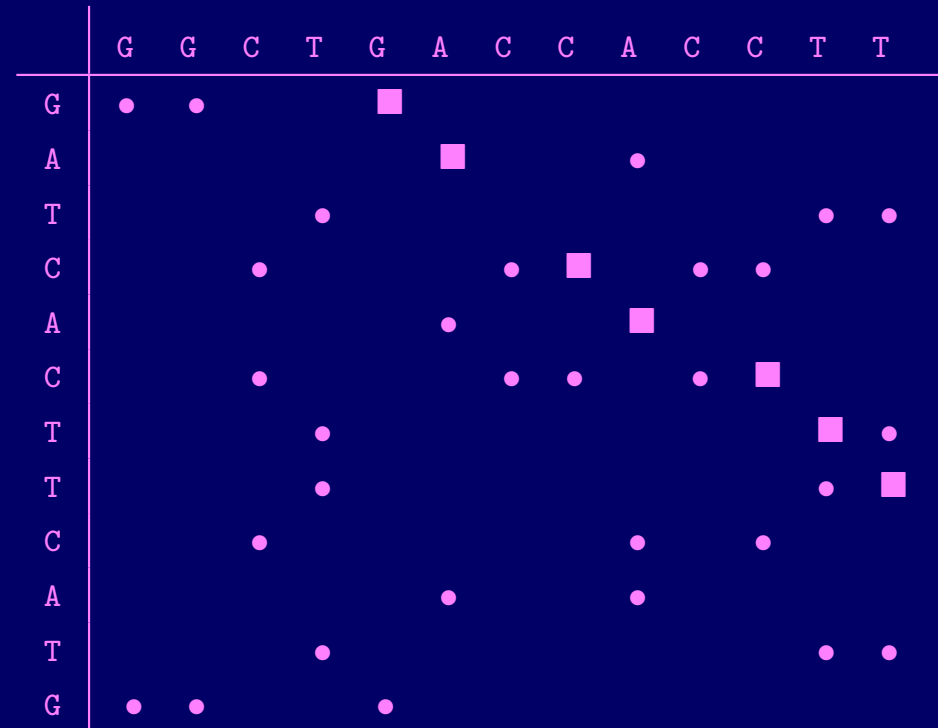
```

G G C T G A C C A C C - T T
|   |   | |   | |   |
G A - T C A C T T C C A T G
    
```

Alignement local

```

G A C C A C C T T
| |   | | |   | |
G A T C A C - T T
    
```



Les séquences présentent une similarité que l'alignement global ne révèle pas.

- s : une matrice de score ;
- g : pénalité associée à un indel ;

- Initialisation :
$$\begin{cases} M(0, 0) = 0 \\ M(0, j) = 0 \quad \leftarrow - \\ M(i, 0) = 0 \quad \leftarrow - \end{cases}$$

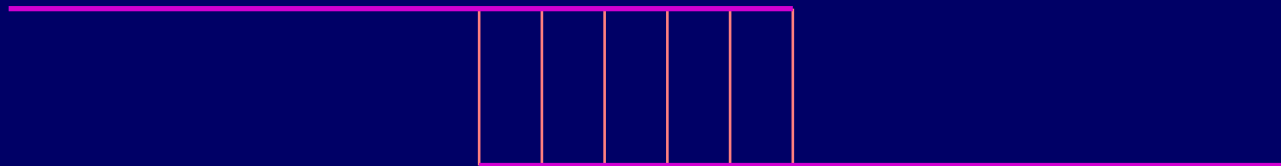
- Remplissage :

$$M(i, j) = \max \begin{cases} M(i-1, j-1) + s(x_i, y_j) & \text{Sub. ou App. exact} \\ M(i-1, j) + g & \text{Délétion} \\ M(i, j-1) + g & \text{Insertion} \\ 0 & \leftarrow - \end{cases}$$

Voir feuille de TD

- Ne pénalise pas les gaps aux extrémités
- Sinon, similaire à l'alignement global
- Permet de détecter des similarités de type **inclusion** et **chevauchement**

Exemple chevauchement



C	A	G	C	A	C	T	T	G	G	A	T	T	C	T	C	G
C	A	G	C	-	-	-	-	-	G	-	T	-	-	-	-	G

C	A	G	C	A	-	C	T	T	G	G	A	T	T	C	T	C	G
-	-	-	C	A	G	C	G	T	G	G	-	-	-	-	-	-	-

⇒ L'alignement global préfère le 1^{er} alignement.

- Initialisation \Rightarrow idem à l'alignement local :
$$\left\{ \begin{array}{l} M(0, 0) = 0 \\ M(0, j) = 0 \quad \leftarrow \\ M(i, 0) = 0 \quad \leftarrow \end{array} \right.$$
- Remplissage \Rightarrow idem à l'alignement global
- Meilleur alignement \Rightarrow valeur max dans la dernière ligne et la dernière colonne.

Pénalité associée aux gaps

- Biologie : insertion d'un segment dans un gène
⇒ les insertions/délétions arrivent en groupe
- **gap** : suite de l insertions ou de l délétions
pénalisation moindre que $l \times g$
- Fonctions de pénalité les plus réalistes :
 - fonctions affines : $o + e \times l$
 - o : pénalité d'ouverture de gap
 - e : pénalité d'extension de gap
 - fonctions logarithmiques
- fonctions de gaps différentes ⇒ algorithmes différents

Différents alignements possibles :

o	0	1	1	1
e	0	0	0.1	1

A11	7	5	4.9	4	ATGCGggACaTG AgGCG--cC-TG	(7 id., 1 gap d'1 pb, 1 gap de 2 pb)
A12	7	5	4.9	4	ATGCGGgaCaTG AgGCGc--C-TG	(7 id., 1 gap d'1 pb, 1 gap de 2 pb)
A13	7	6	5.8	4	ATGCGggaCATG AgGCG---CcTG	(7 id., 1 gap de 3 pb)
A14	7	4	4	4	ATGCGgGaCaTG AgGCG-c-C-TG	(7 id., 3 gaps de 1 pb)

- Séquences protéiques d'enzymes dépendantes de la thiamine phosphatase (TPP) : `ILV1_TOBAC` et `ILVB_ARATH`
- Alignement global avec des pénalités 2/2 ($o = 2$ et $e = 2$) et 12/2.
- Logiciel : `stretcher` de la suite `EMBOSS`
- **Résultat** : Scores et pourcentage d'identité sensiblement identiques : 2711 contre 2644, 77.3% contre 76.6% ; mais l'alignement avec pénalités 12/2 ne contient qu'un gap contre 22 avec les pénalités 2/2.
- Remarque : le score le plus élevé n'est pas nécessairement celui du meilleur alignement

Fonction : $c(g) = -o - (g - 1) \times e$

Match ou mismatch :

$$M(i, j) = \max \begin{cases} M(i - 1, j - 1) + s(x_i, y_j) \\ D(i - 1, j - 1) + s(x_i, y_j) \\ I(i - 1, j - 1) + s(x_i, y_j) \end{cases}$$

Délétion :

$$D(i, j) = \max \begin{cases} M(i - 1, j) - o \\ D(i - 1, j) - e \end{cases}$$

Insertion :

$$I(i, j) = \max \begin{cases} M(i, j - 1) - o \\ I(i, j - 1) - e \end{cases}$$

Voir feuille de TD

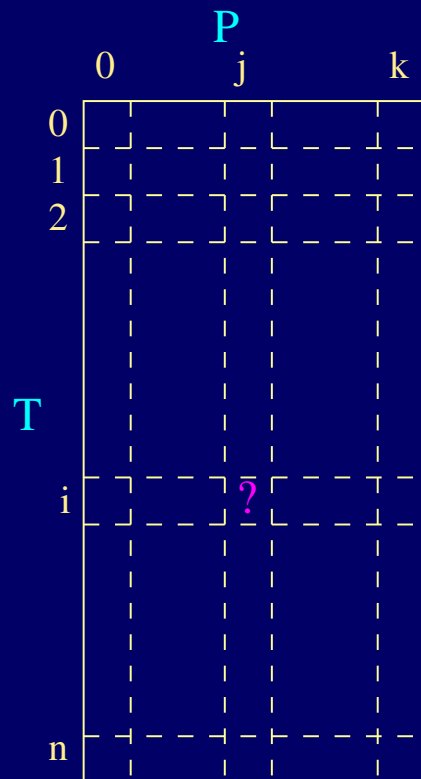
- *Wraparound Dynamic Programming (WDP)* : alignement d'une séquence T avec une répétition en tandem de motif P aussi longue que nécessaire
- Matrice de programmation dynamique de taille $O(|T| \times |P|)$.
- Algorithme de [Myers et Miller 89] et [Fischetti et al. 92]

- Séquence T de longueur n ; Motif P de longueur k ;

-
- Séquence T de longueur n ; Motif P de longueur k ;
 - Matrice \mathcal{M} de dimension $(n + 1) \times (k + 1)$;

- Séquence T de longueur n ; Motif P de longueur k ;
- Matrice \mathcal{M} de dimension $(n + 1) \times (k + 1)$;
- $\mathcal{M}(i, j)$ représente le score optimal d'alignement entre $T[1..i]$ et $P^*P[1..j]$;

- Séquence T de longueur n ; Motif P de longueur k ;
- Matrice \mathcal{M} de dimension $(n + 1) \times (k + 1)$;
- $\mathcal{M}(i, j)$ représente le score optimal d'alignement entre $T[1..i]$ et $P^*P[1..j]$;



- Principe de l'algorithme :

→ Initialisation :

$$\begin{cases} \mathcal{M}(0, 0) = 0 \\ \mathcal{M}(i, 0) = i \times Del \\ \mathcal{M}(0, j) = j \times Ins \end{cases}$$

- Récurrence générale : 2 passes par ligne.

- 1^{re} passe :

$$\mathcal{M}(i, 1) = \min \left\{ \begin{array}{ll} \mathcal{M}(i-1, 0) + \textit{Sub}(T[i], P[1]) & \textit{Sub}(T[i], P[1]) \\ \mathcal{M}(i-1, k) + \textit{Sub}(T[i], P[1]) & \textit{WSub}(T[i], P[1]) \\ \mathcal{M}(i-1, 1) + \textit{Del} & \textit{Délétion de } T[i] \\ \mathcal{M}(i, 0) + \textit{Ins} & \textit{Insertion de } P[1] \end{array} \right.$$

$$\mathcal{M}(i, j) = \min \left\{ \begin{array}{ll} \mathcal{M}(i-1, j-1) + \textit{Sub}(T[i], P[j]) & \textit{Sub}(T[i], P[j]) \\ \mathcal{M}(i-1, j) + \textit{Del} & \textit{Délétion de } T[i] \\ \mathcal{M}(i, j-1) + \textit{Ins} & \textit{Insertion de } P[j] \end{array} \right.$$

- 1^{re} passe :

$$\mathcal{M}(i, 1) = \min \left\{ \begin{array}{ll} \mathcal{M}(i-1, 0) + \textit{Sub}(T[i], P[1]) & \textit{Sub}(T[i], P[1]) \\ \mathcal{M}(i-1, k) + \textit{Sub}(T[i], P[1]) & \textit{WSub}(T[i], P[1]) \\ \mathcal{M}(i-1, 1) + \textit{Del} & \textit{Délétion de } T[i] \\ \mathcal{M}(i, 0) + \textit{Ins} & \textit{Insertion de } P[1] \end{array} \right.$$

$$\mathcal{M}(i, j) = \min \left\{ \begin{array}{ll} \mathcal{M}(i-1, j-1) + \textit{Sub}(T[i], P[j]) & \textit{Sub}(T[i], P[j]) \\ \mathcal{M}(i-1, j) + \textit{Del} & \textit{Délétion de } T[i] \\ \mathcal{M}(i, j-1) + \textit{Ins} & \textit{Insertion de } P[j] \end{array} \right.$$

- 2^e passe :

$$\mathcal{M}(i, 1) = \min \left\{ \begin{array}{ll} \mathcal{M}(i, 1) & \textit{Valeur précédente} \\ \mathcal{M}(i, k) + \textit{Ins} & \textit{W. Insertion de } P[1] \end{array} \right.$$

$$\mathcal{M}(i, j) = \min \left\{ \begin{array}{ll} \mathcal{M}(i, j) & \textit{Valeur précédente} \\ \mathcal{M}(i, j-1) + \textit{Ins} & \textit{Insertion de } P[j] \end{array} \right.$$

Aligner $T=CTCTAGC$ avec une répétition en tandem du motif $P=ACT$.

avec $Ins = Del = 2$ et $Sub(a, b) = 3$ si $a \neq b$ et 0 sinon.

		A	C	T
	0	2	4	6
C	2	3	2	4
T	4	4	4	2
C	6	5	4	4
T	8	6	6	4
A	10	4	6	6
G	12	6	7	8
C	14	8	6	8

- C T - C T A G C -
 | | | | |
 A C T A C T A - C T

-
- Introduction
 - Dotplot
 - Alignement : définitions et scores
 - Algorithme d'alignement (programmation dynamique)
 - Autres types d'alignements
 - La suite ...

- Jusque ici les algorithmes considérés donnent des solutions exactes mais ils ne sont pas très rapides (complexité $O(n \times m)$)

~ 100 millions de résidus dans les BD

Comparer une séq. de 1000pb à une BD $\Rightarrow \sim 10^{11}$ cellules à évaluer

Calcul 10 millions de cellules par sec. $\Rightarrow 10^4$ sec. = ~ 3 heures

\Rightarrow Nécessité d'algorithmes plus rapides

- Algorithmes heuristiques, les plus connus : BLAST et FASTA

Comment peut-on savoir d'après le score d'alignement si les séquences se ressemblent ?

- Tout alignement produit un score

→ *Quel serait le score de l'alignement de 2 séquences aléatoires ?*

- **Approche empirique** : Tests de permutation (TP logiciel PRSS)
- **Approche statistique** : **E-value** = Nombre attendu de fois de trouver un alignement de score supérieur à S par hasard quand on aligne une séquence de longueur m avec une séquence de longueur n