M1 Bioinformatique, Connaissances et Données

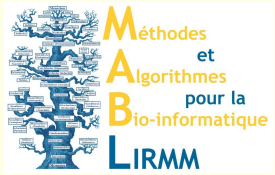
Master Sciences et Numérique pour la Santé

Année 2017-2018

HMSN206 - Partie Alignement

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

Anne-Muriel Chifolleau



LIRMM - UM

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

Qu'est-ce que aligner des séquences ?

Aligner = comparer

- Identifier les points communs ou les différences entre des séquences
- Retracer l'histoire évolutive des séquences en simulant les mutations
  - → Mutations = dégradations ou des changements permanents de l'ADN
    - ▷ Substitution : remplacement d'un nucléotide par un autre
    - ▷ Insertion / Délétion : ajout / suppression d'un fragment d'ADN
    - ▷ Duplication : doublement d'un fragment d'ADN
    - ▷ Recombinaison : échange de fragments d'ADN entre chromosomes ou génomes
    - ▷ Inversion : changement de sens d'un fragment d'ADN
  - → Mutations plus ou moins graves : neutre, défavorable, bénéfique, létale

Pourquoi aligner des séquences ?

- Recherche d'homologie
- Identification de séquences / prédiction de gènes
- Recherche de fonctions ou domaines communs / transfert d'annotation
- Identification de régions répétées / introns / exons
- Assemblage de gènome

On dispose de beaucoup de séquences $\Rightarrow$ ressource à exploiter

Exemples d'alignement : insuline

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)

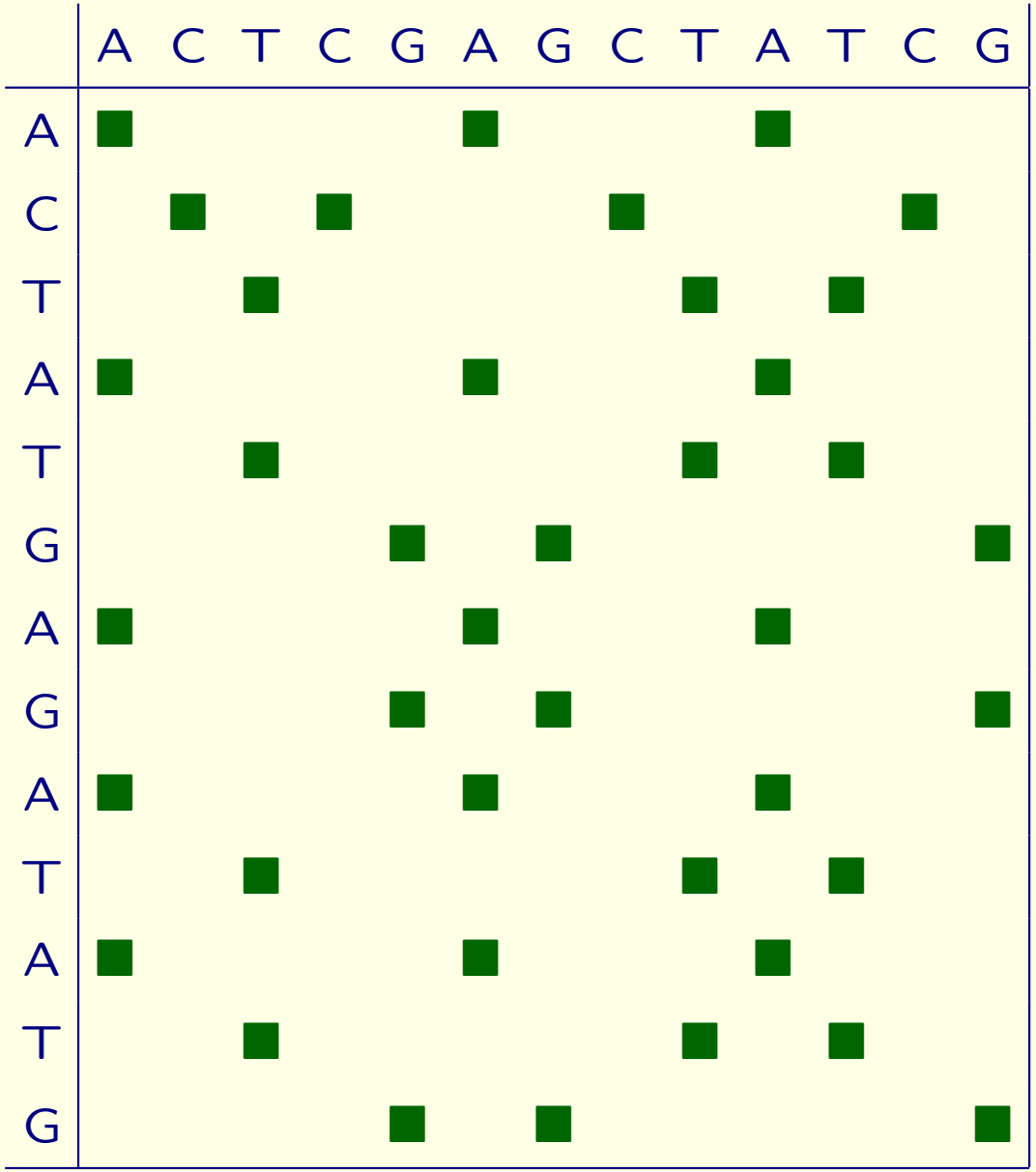
- Introduction
- Dotplot
- Alignement : définitions et scores
- Algorithme d'alignement (programmation dynamique)
- Autres types d'alignements
- Significativité
- Les logiciels
- La suite ...

Une méthode graphique : le Dot Plot

match (identité) → ■

mismatch → □

diagonale = région similaire

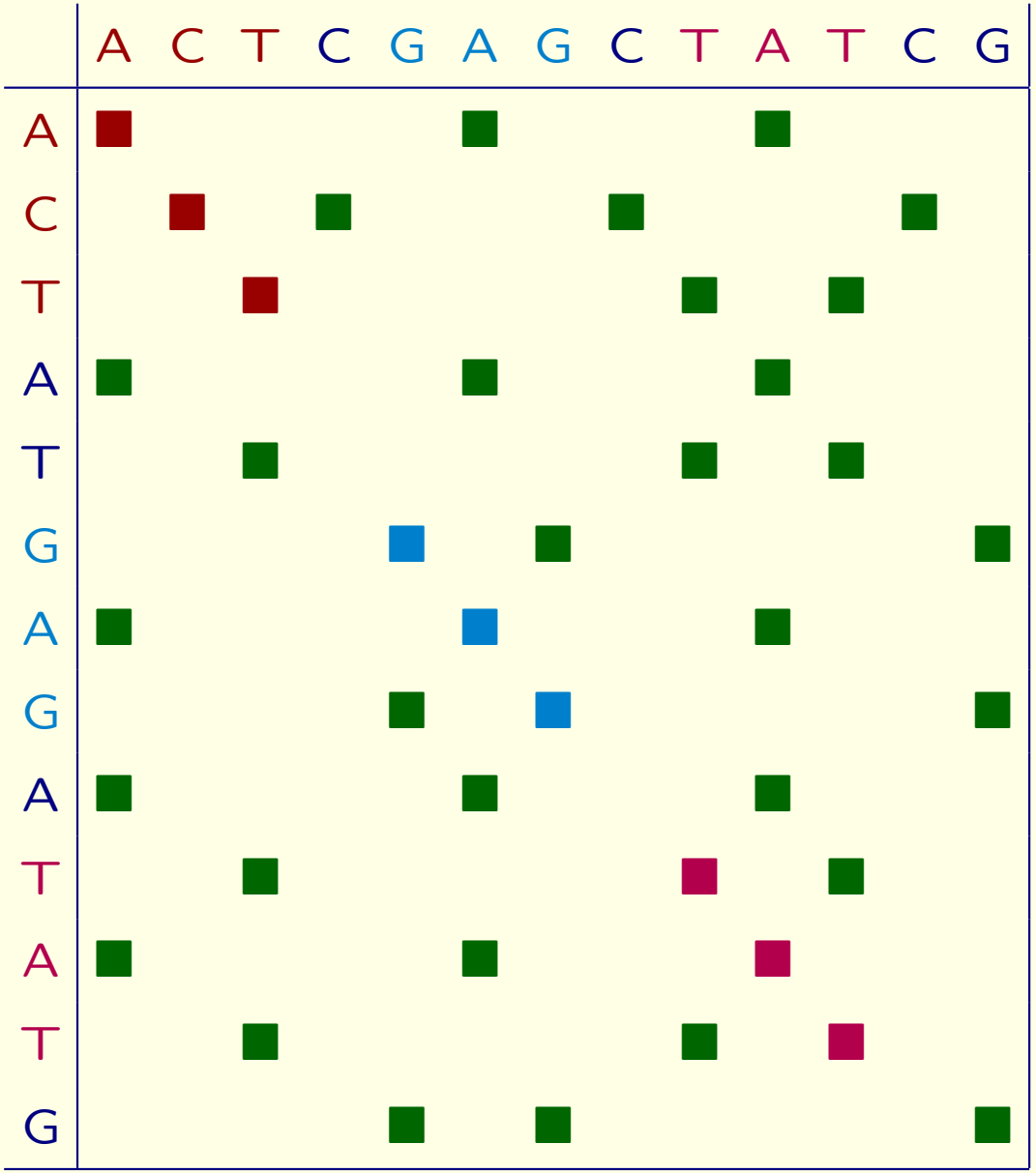


Une méthode graphique : le Dot Plot

match (identité) → ■

mismatch → □

diagonale = région similaire

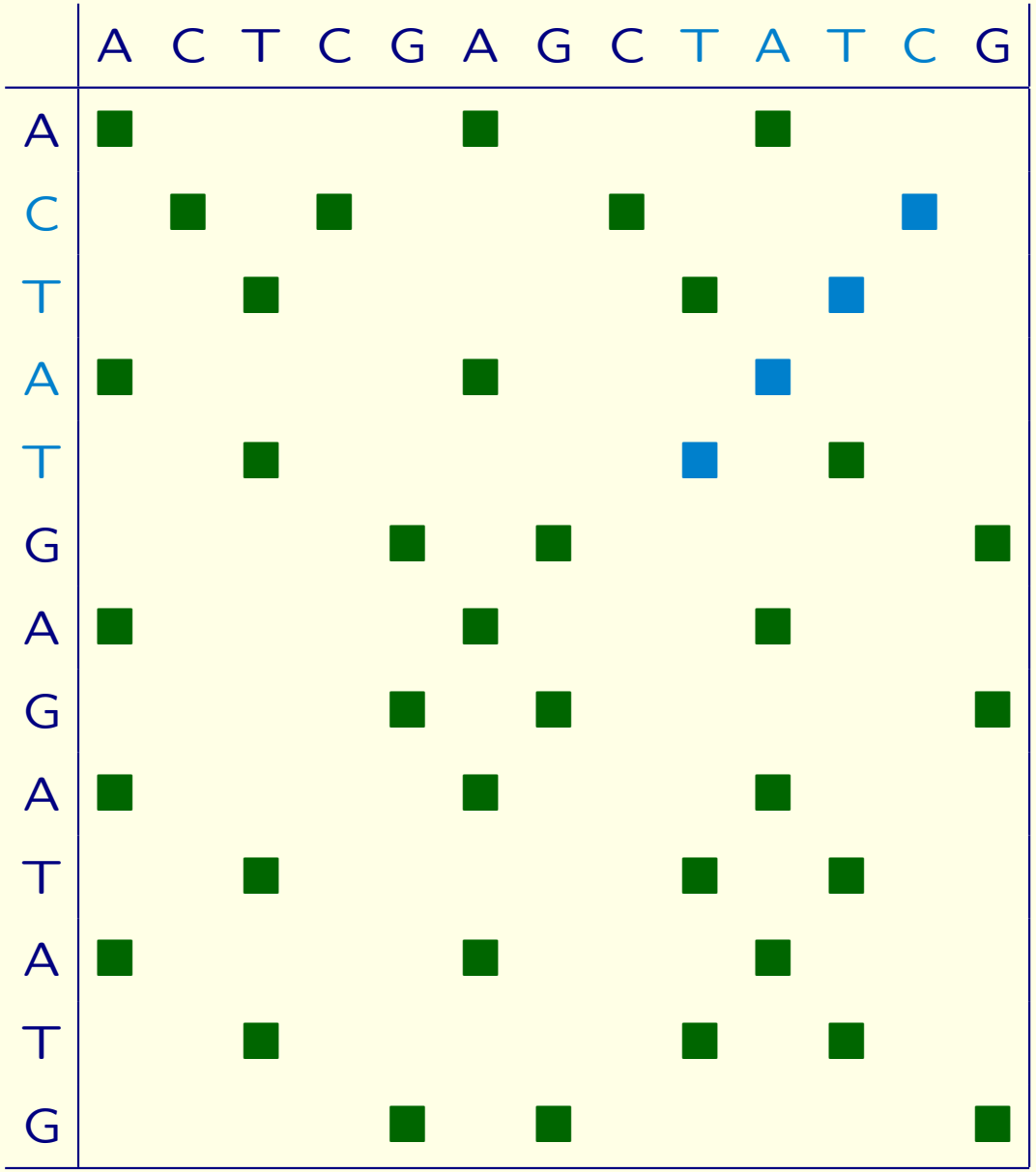


Une méthode graphique : le Dot Plot

match (identité) → ■

mismatch → □

diagonale inversée = région miroir

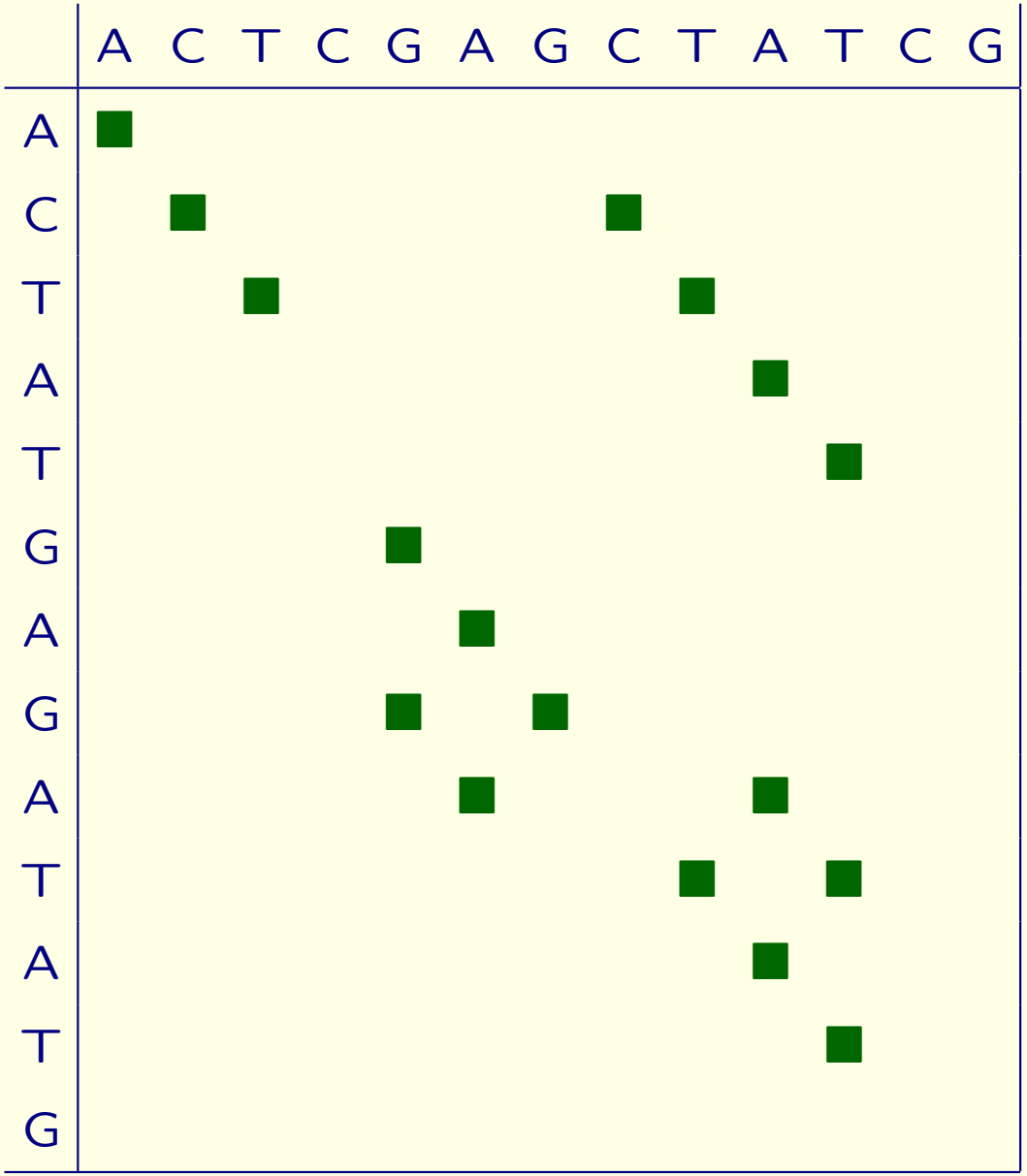


Dot Plot : vue par fenêtre

match (identité) → ■

mismatch → □

Fenêtre de taille 2

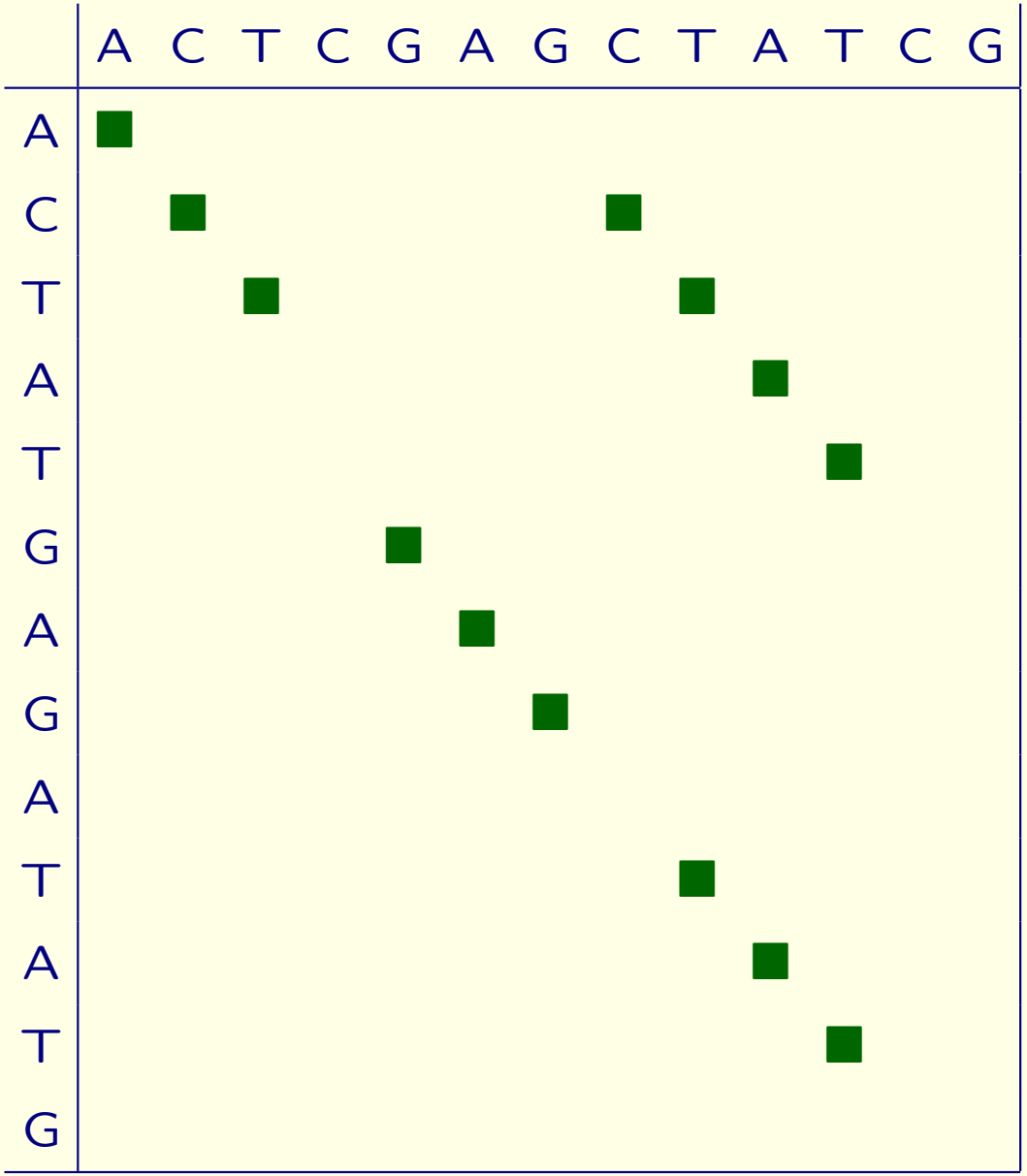


Dot Plot : vue par fenêtre

match (identité) → ■

mismatch → □

Fenêtre de taille 3

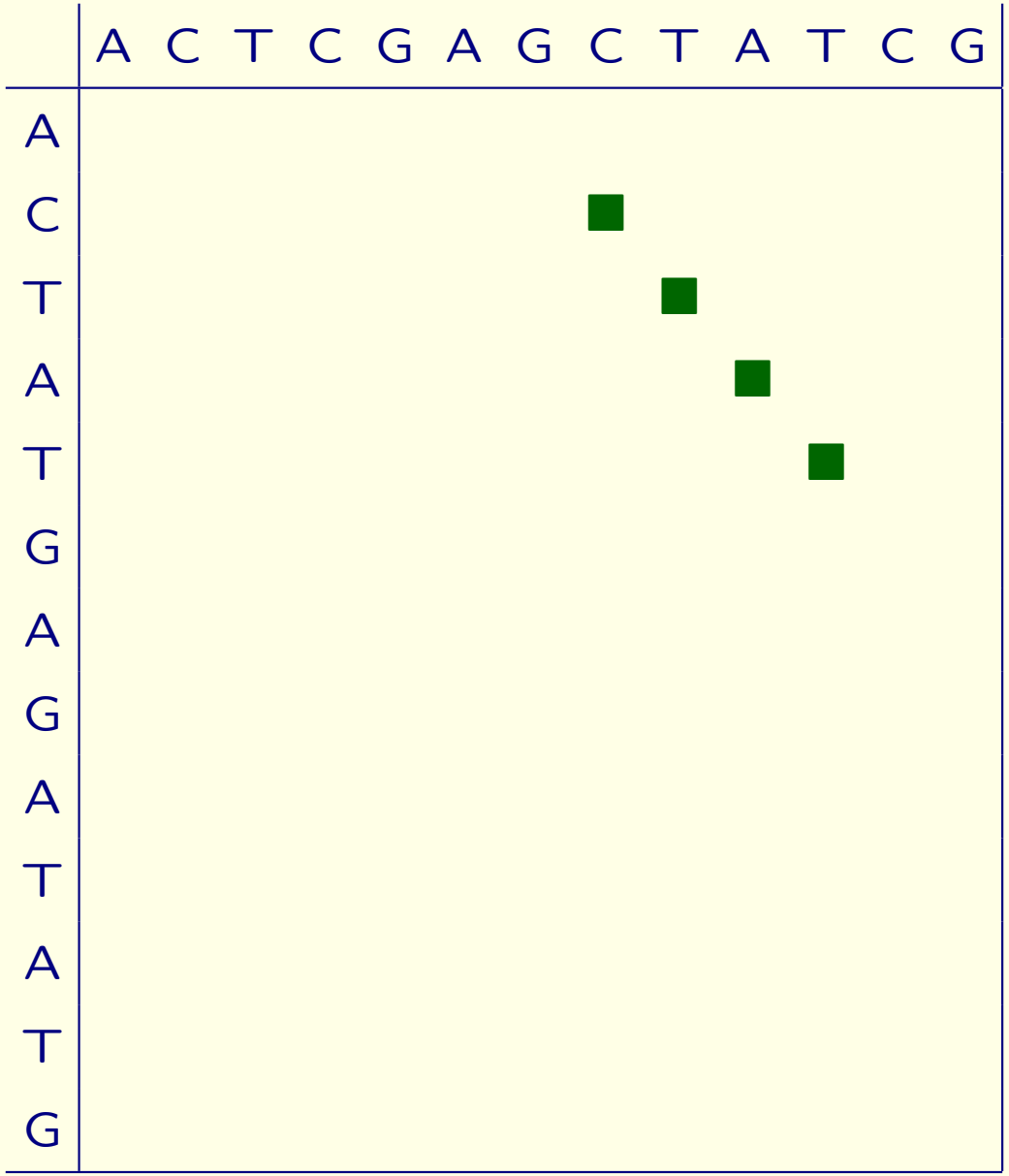


Dot Plot : vue par fenêtre

match (identité) → ■

mismatch → □

Fenêtre de taille 4

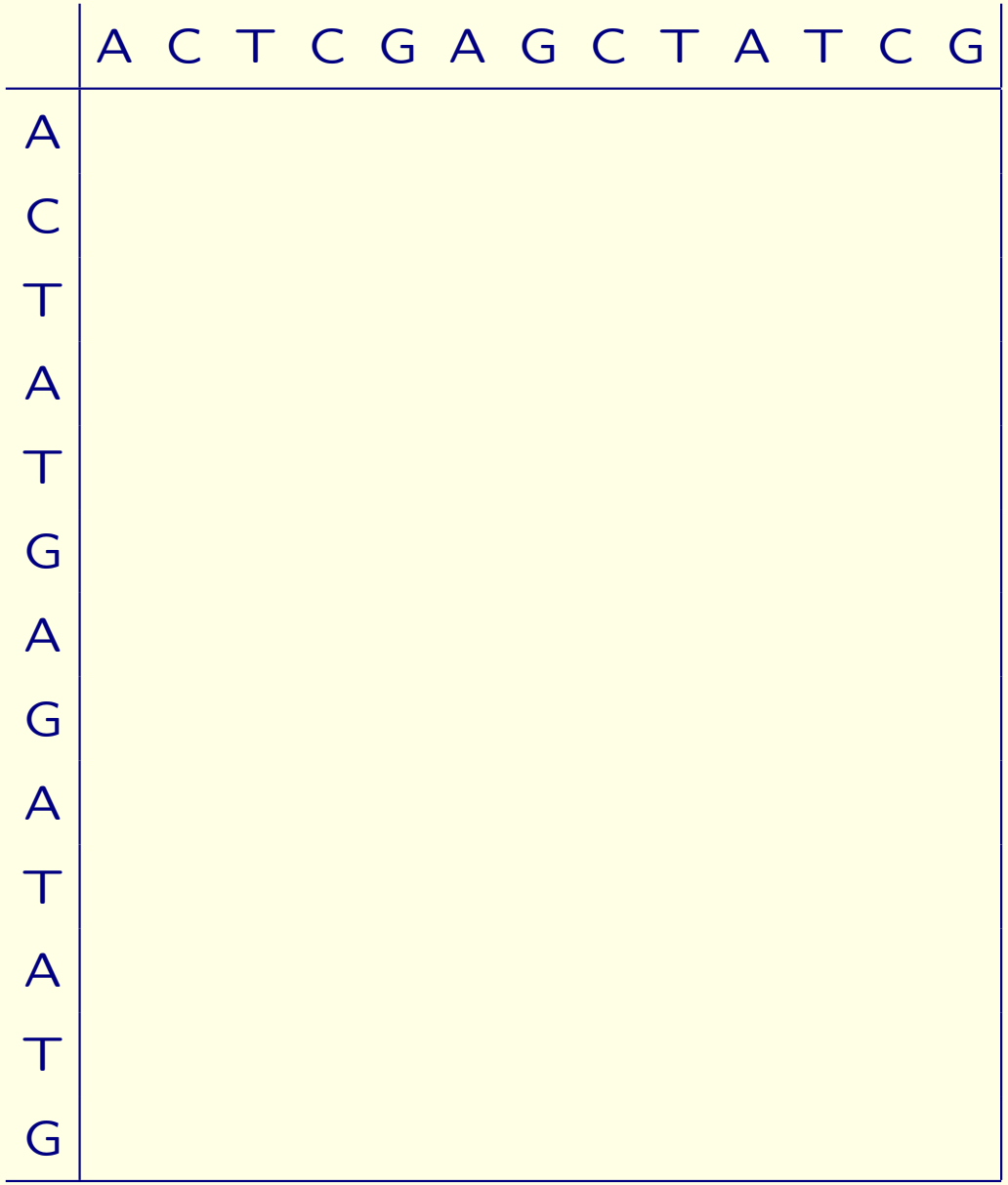


Dot Plot : vue par fenêtre

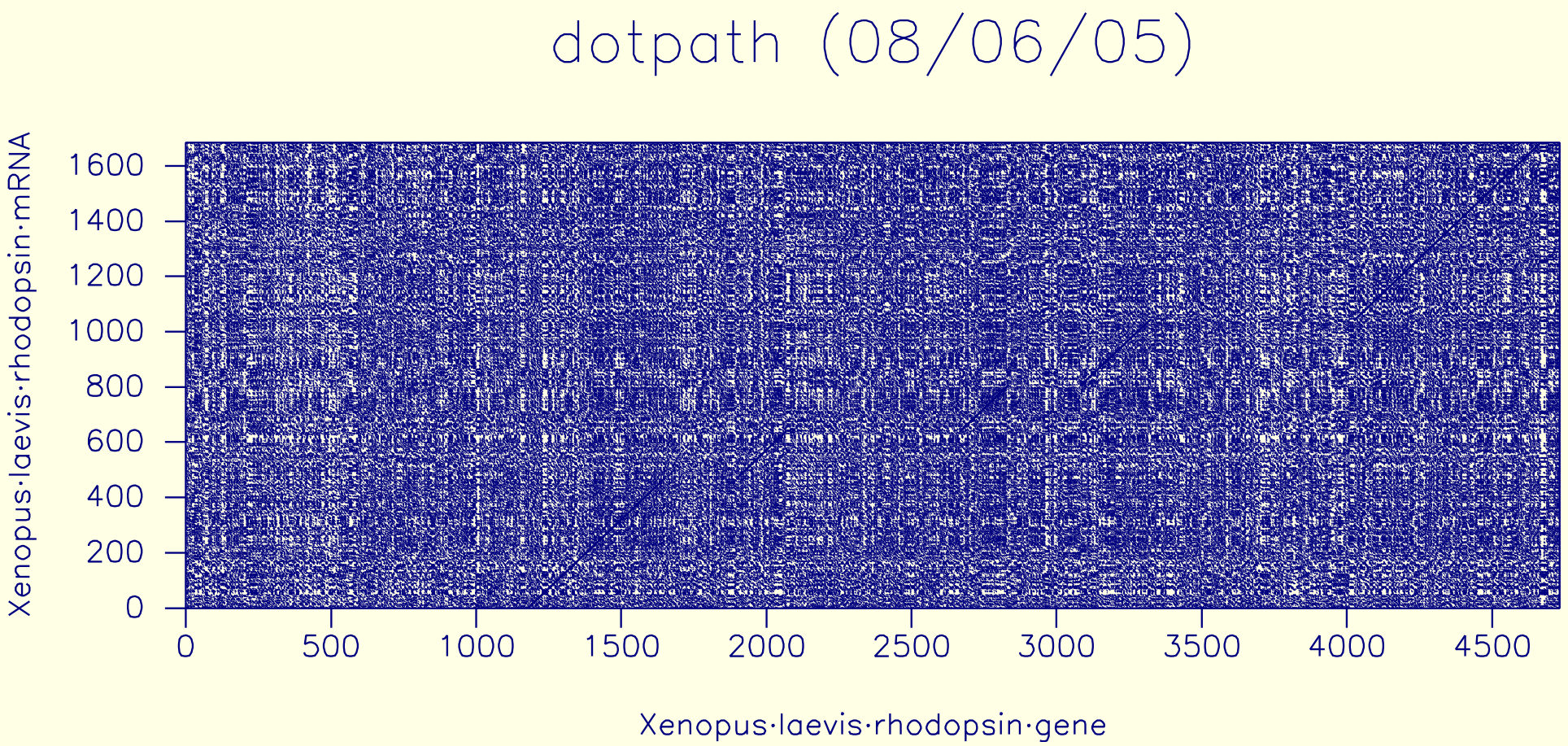
match (identité) → ■

mismatch → □

Fenêtre de taille 5

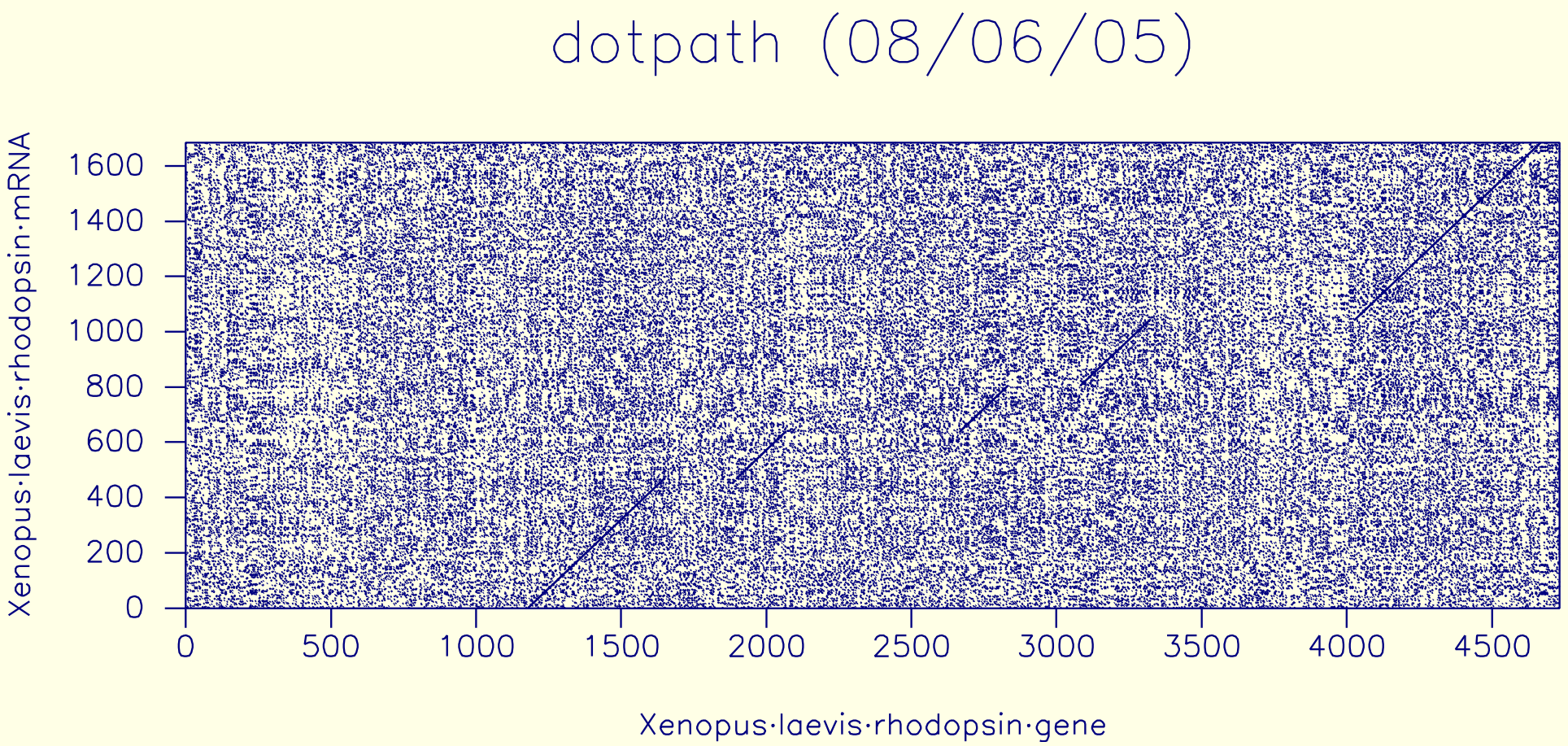


Dotplot sur des séquences réelles



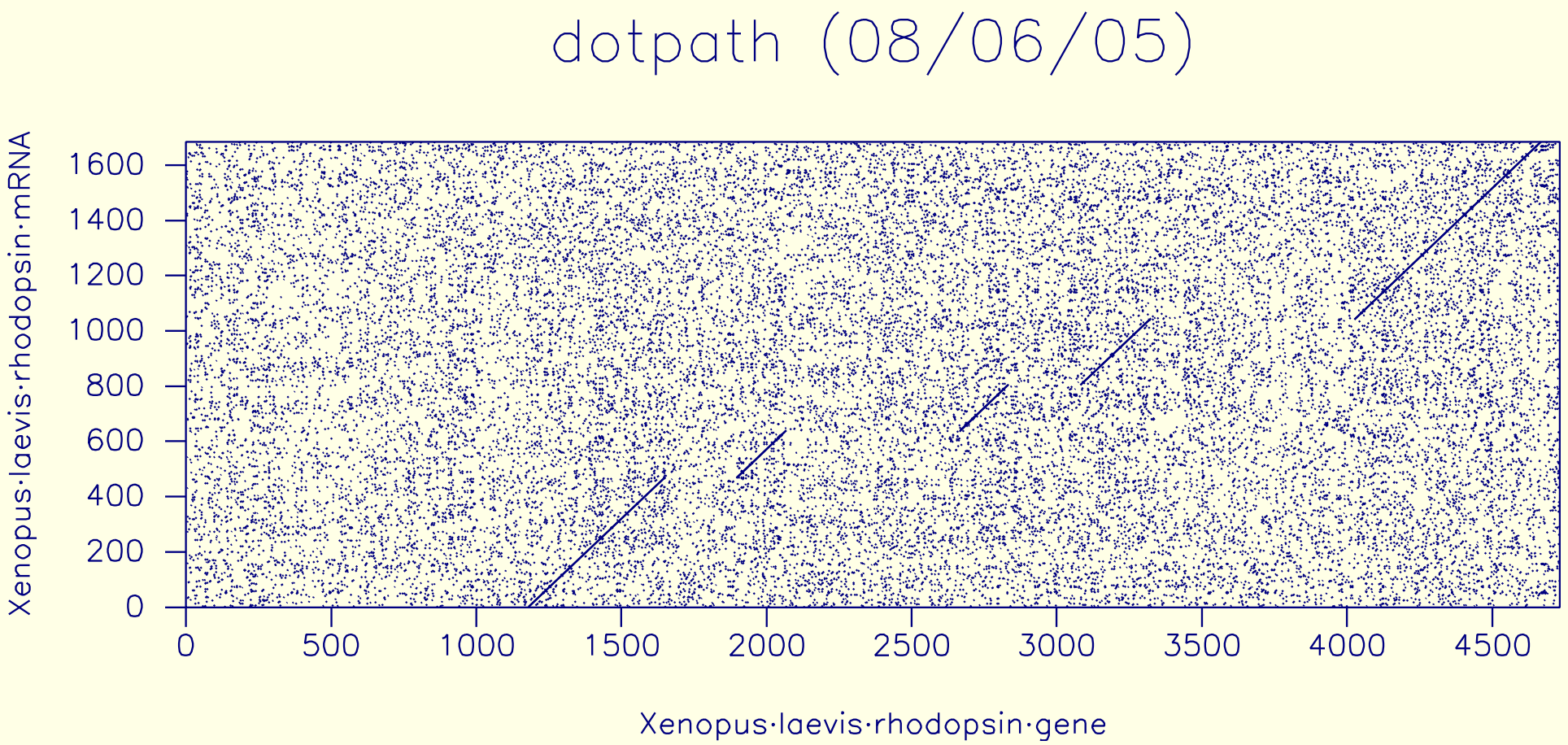
Taille fenêtre = 2

Dotplot sur des séquences réelles



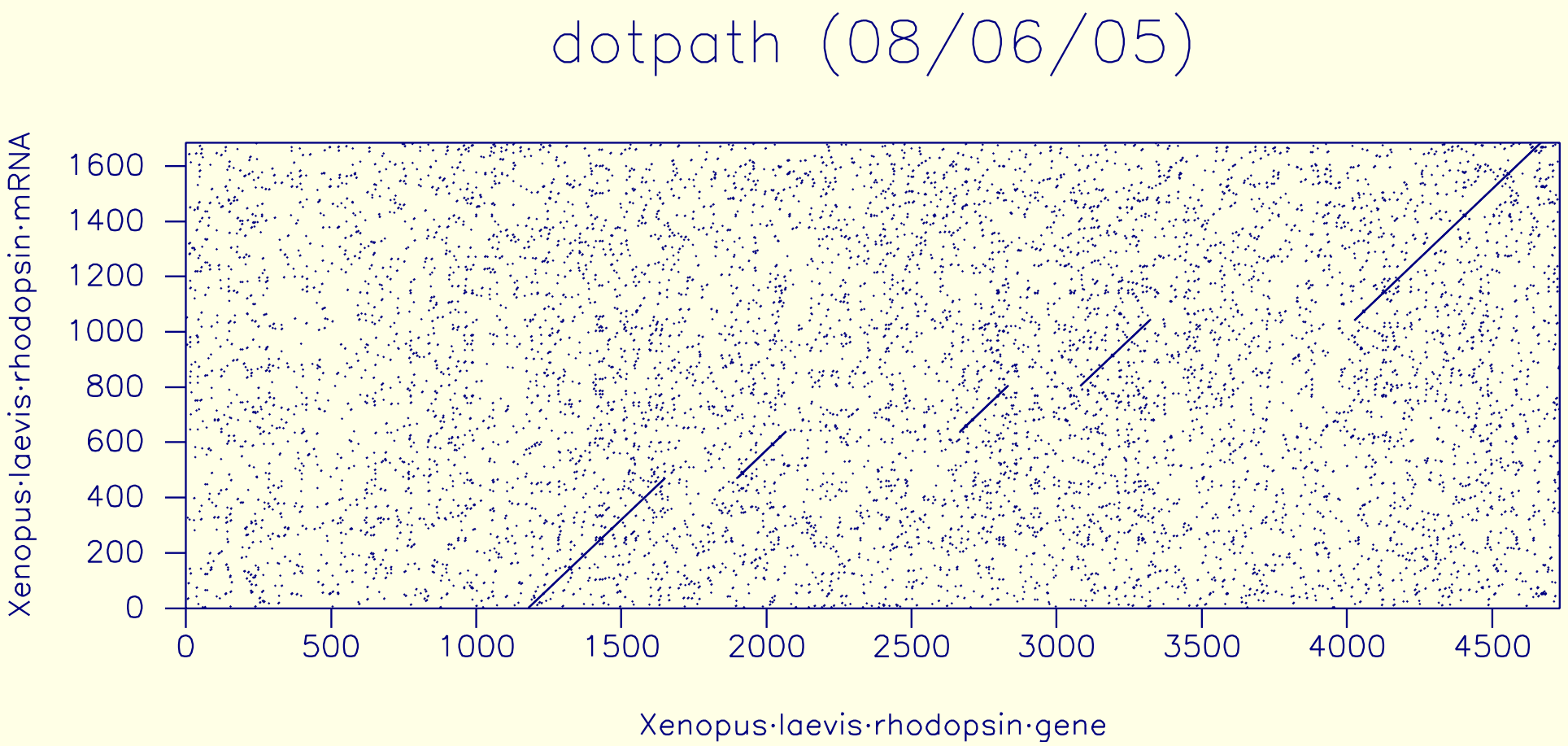
Taille fenêtre = 3

Dotplot sur des séquences réelles



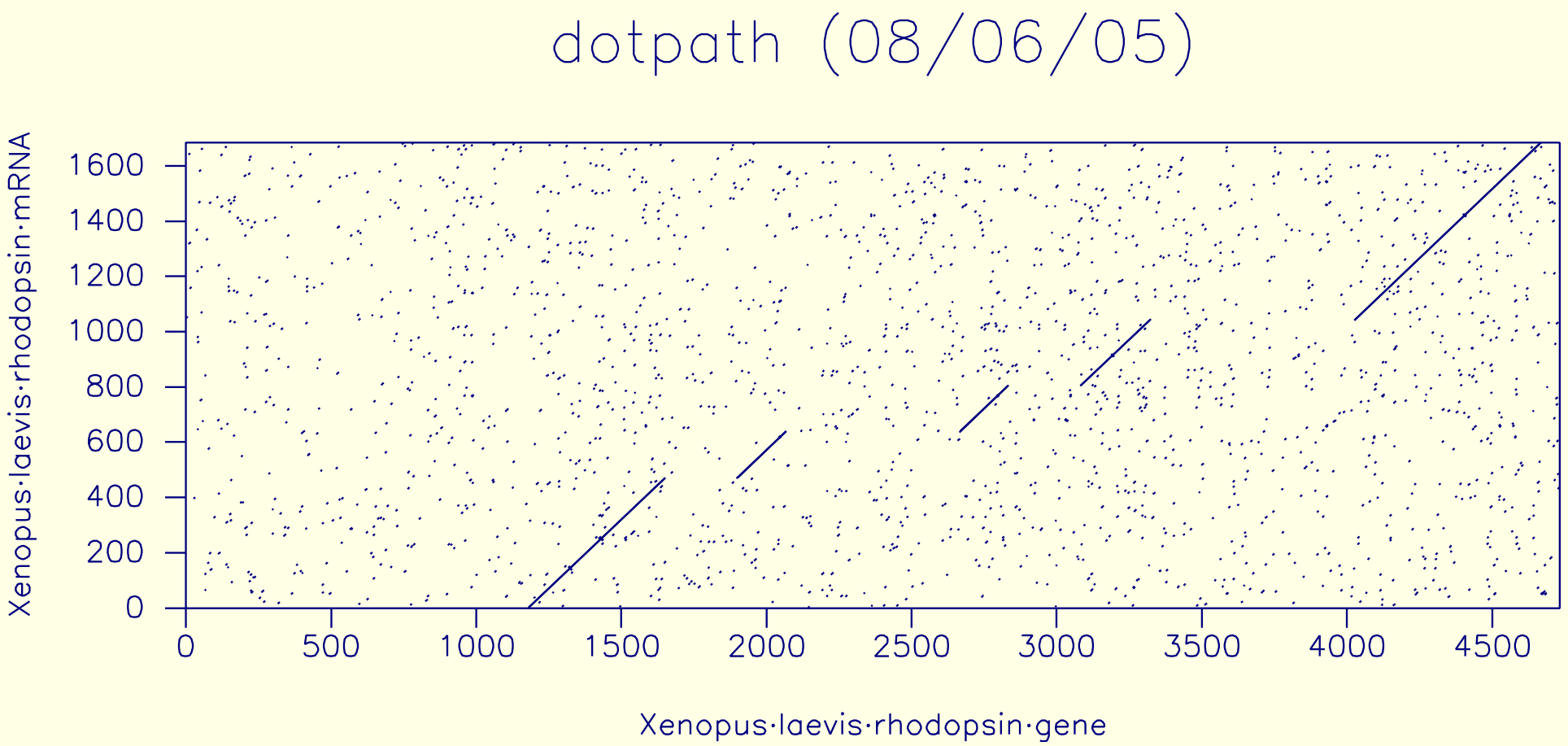
Taille fenêtre = 4

Dotplot sur des séquences réelles



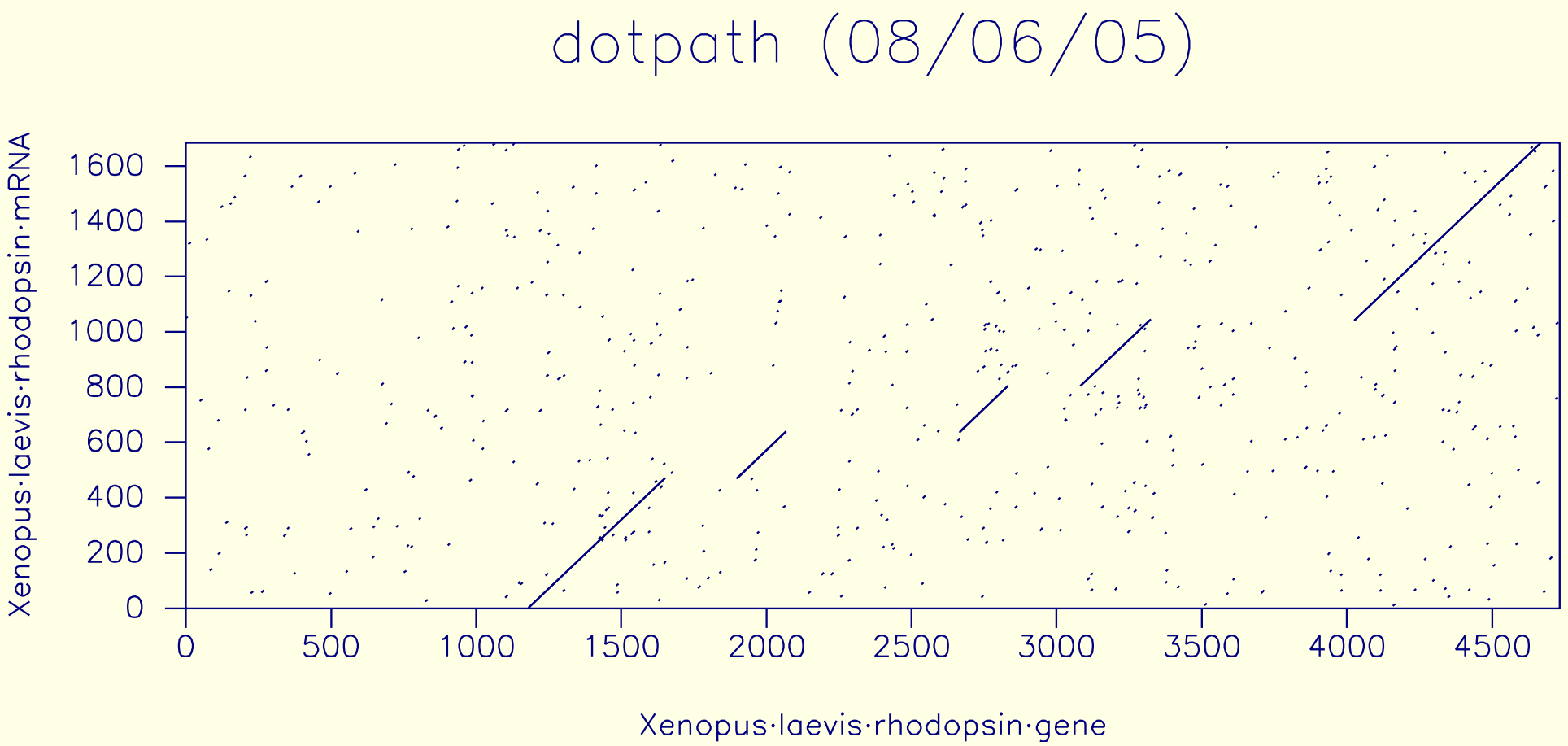
Taille fenêtre = 5

Dotplot sur des séquences réelles



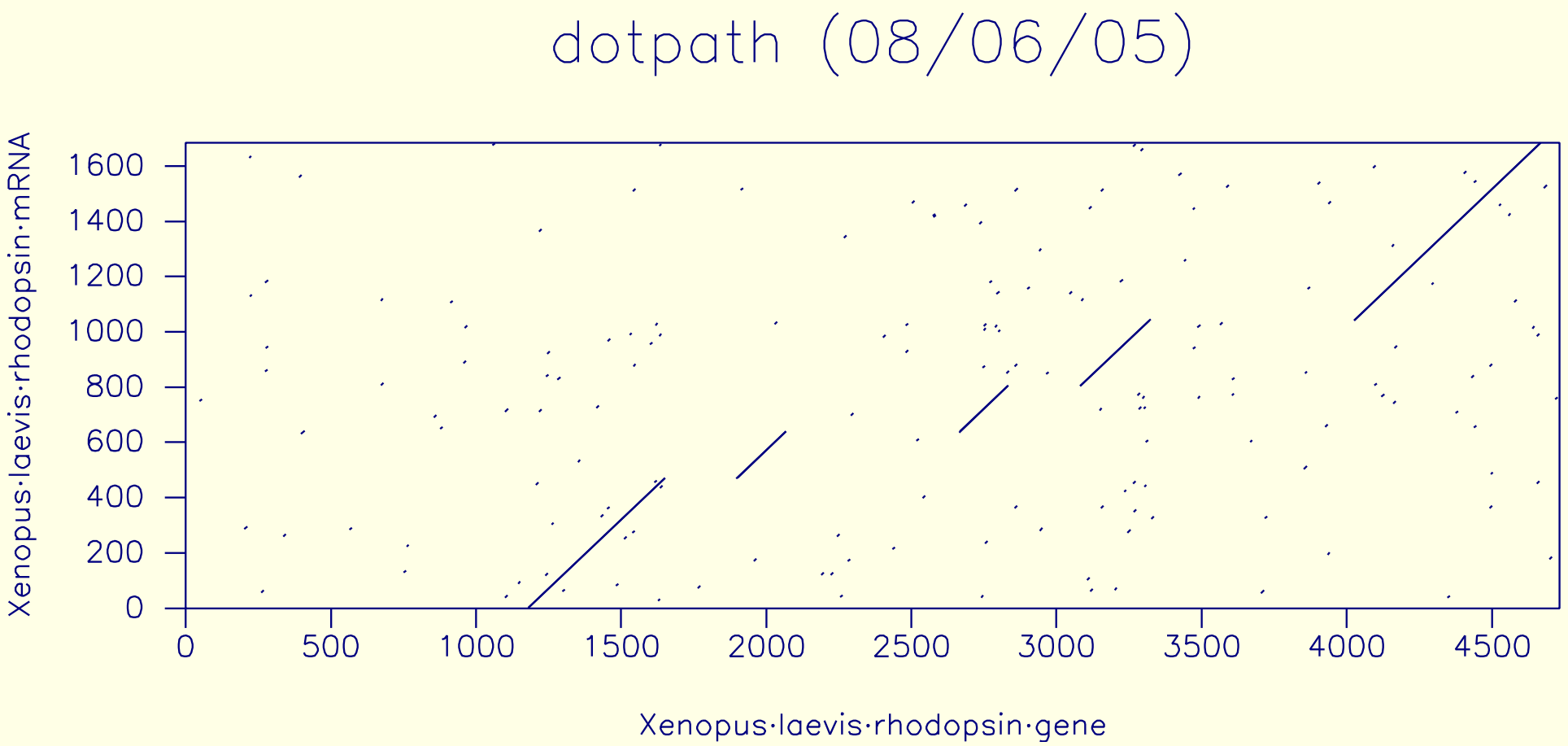
Taille fenêtre = 6

Dotplot sur des séquences réelles



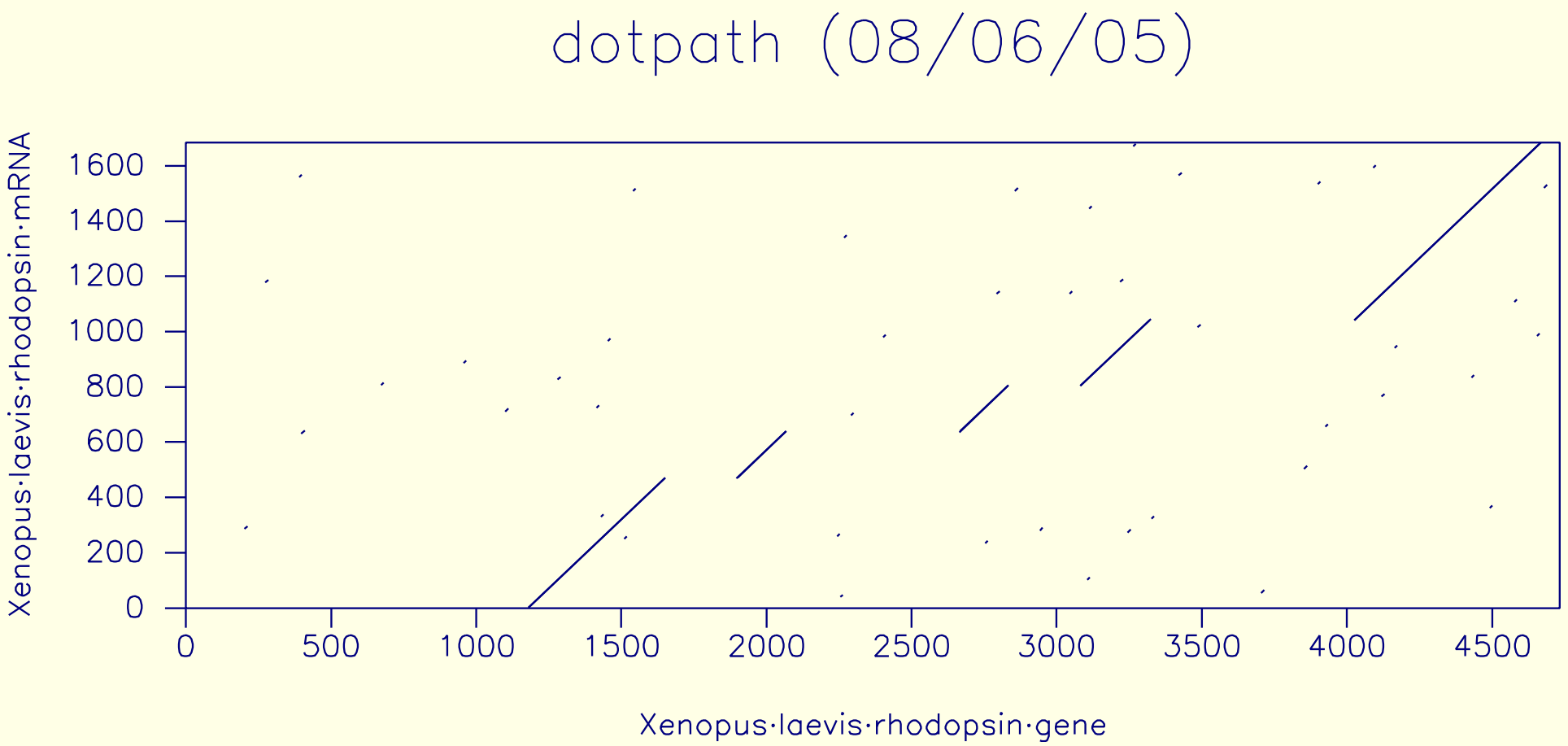
Taille fenêtre = 7

Dotplot sur des séquences réelles



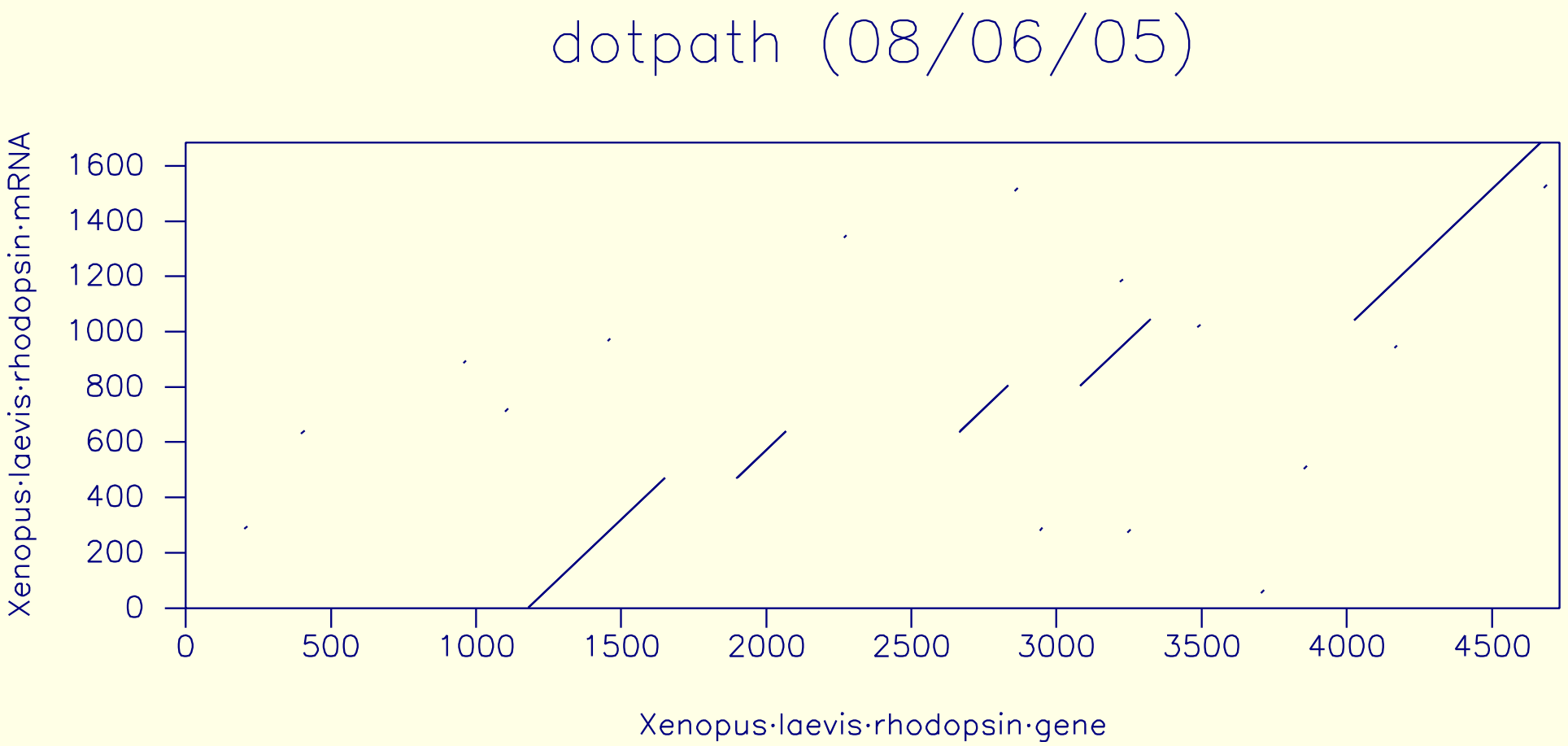
Taille fenêtre = 8

Dotplot sur des séquences réelles



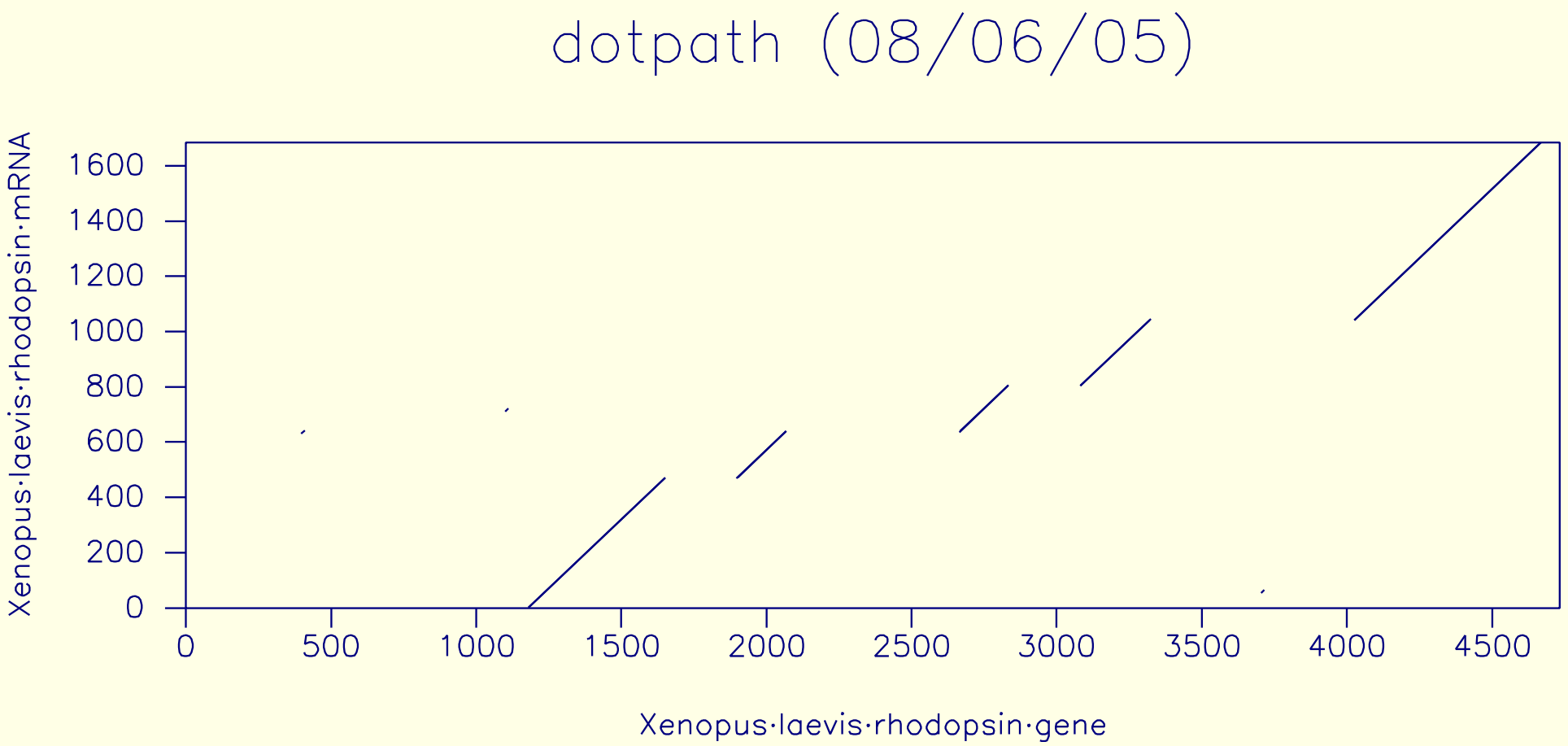
Taille fenêtre = 9

Dotplot sur des séquences réelles



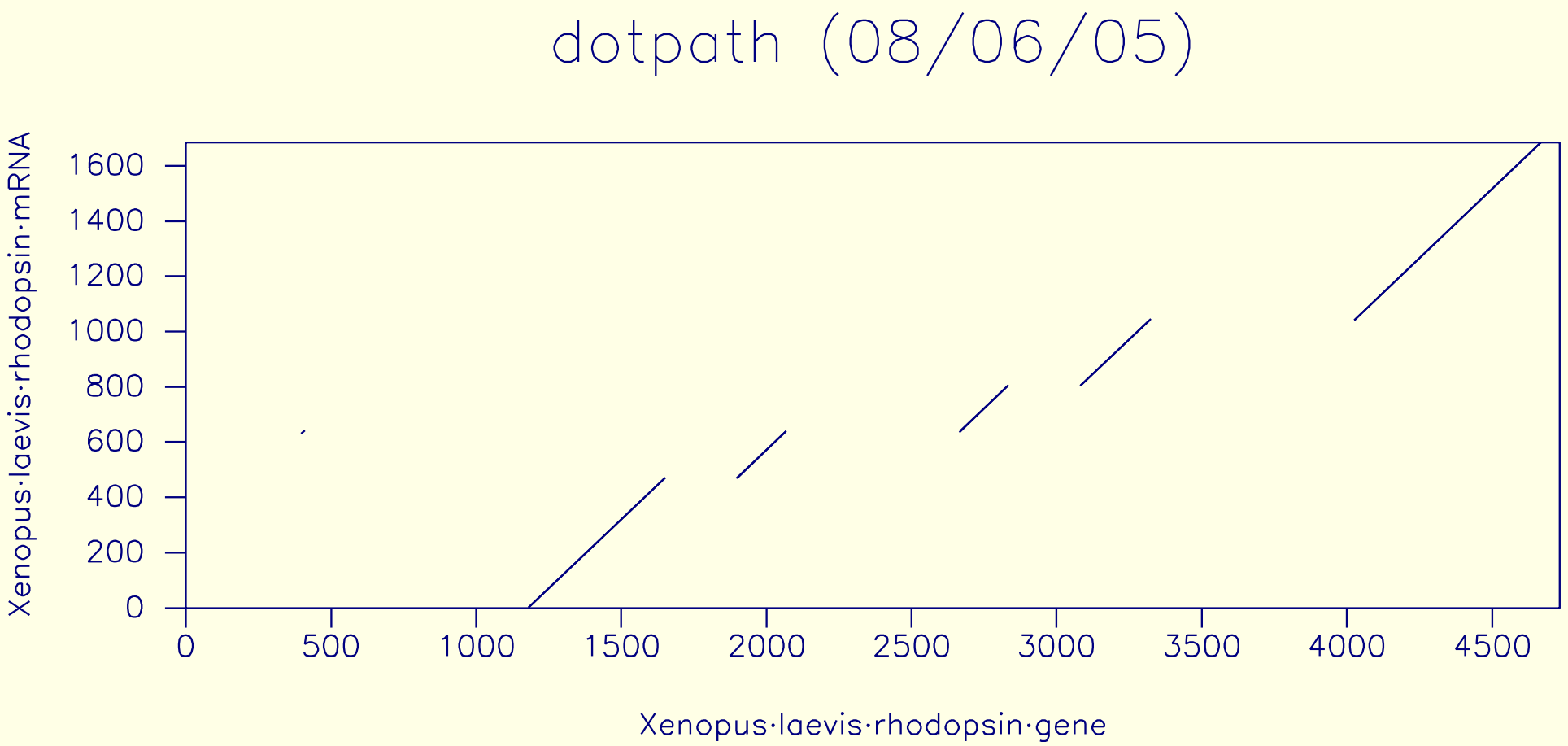
Taille fenêtre = 10

Dotplot sur des séquences réelles



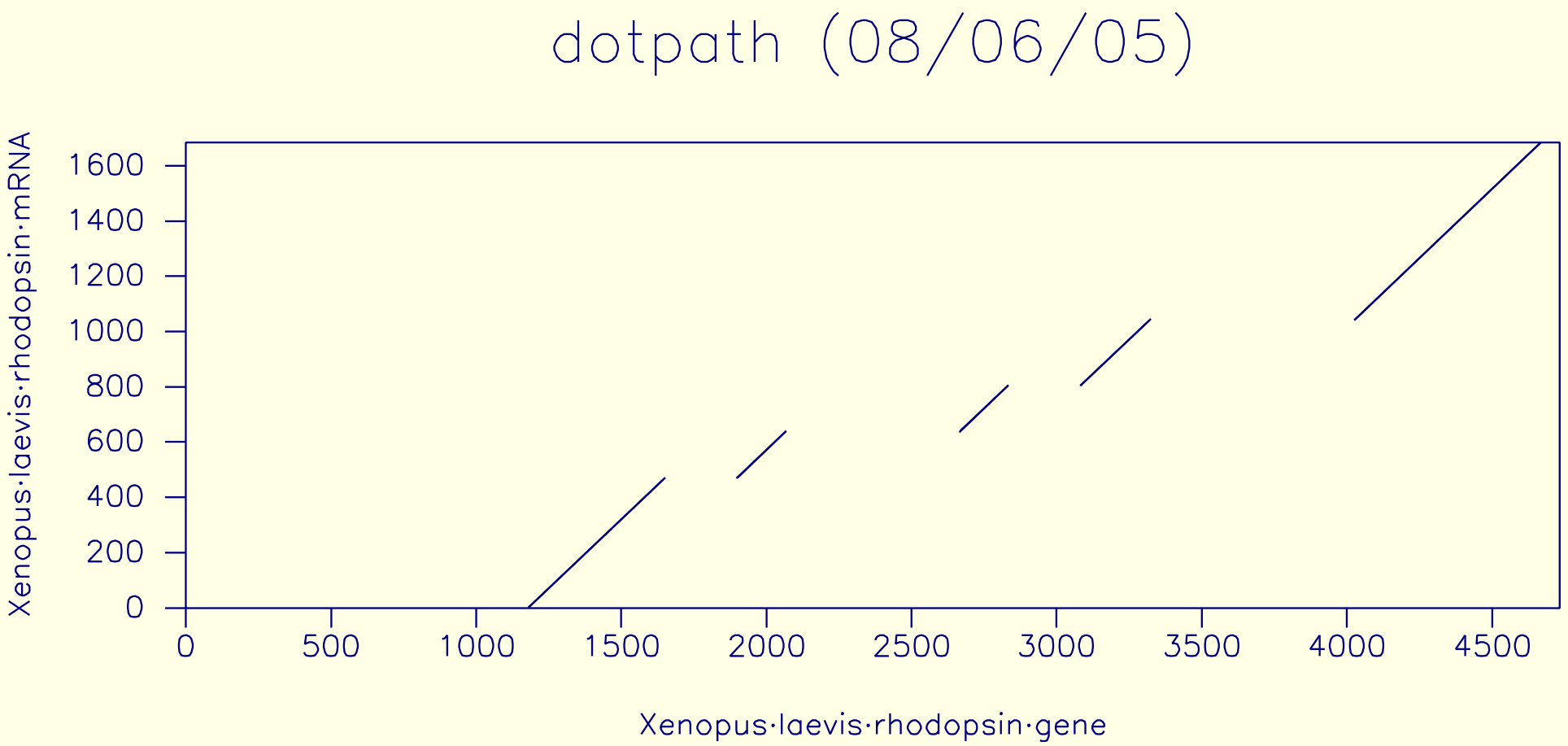
Taille fenêtre = 11

Dotplot sur des séquences réelles



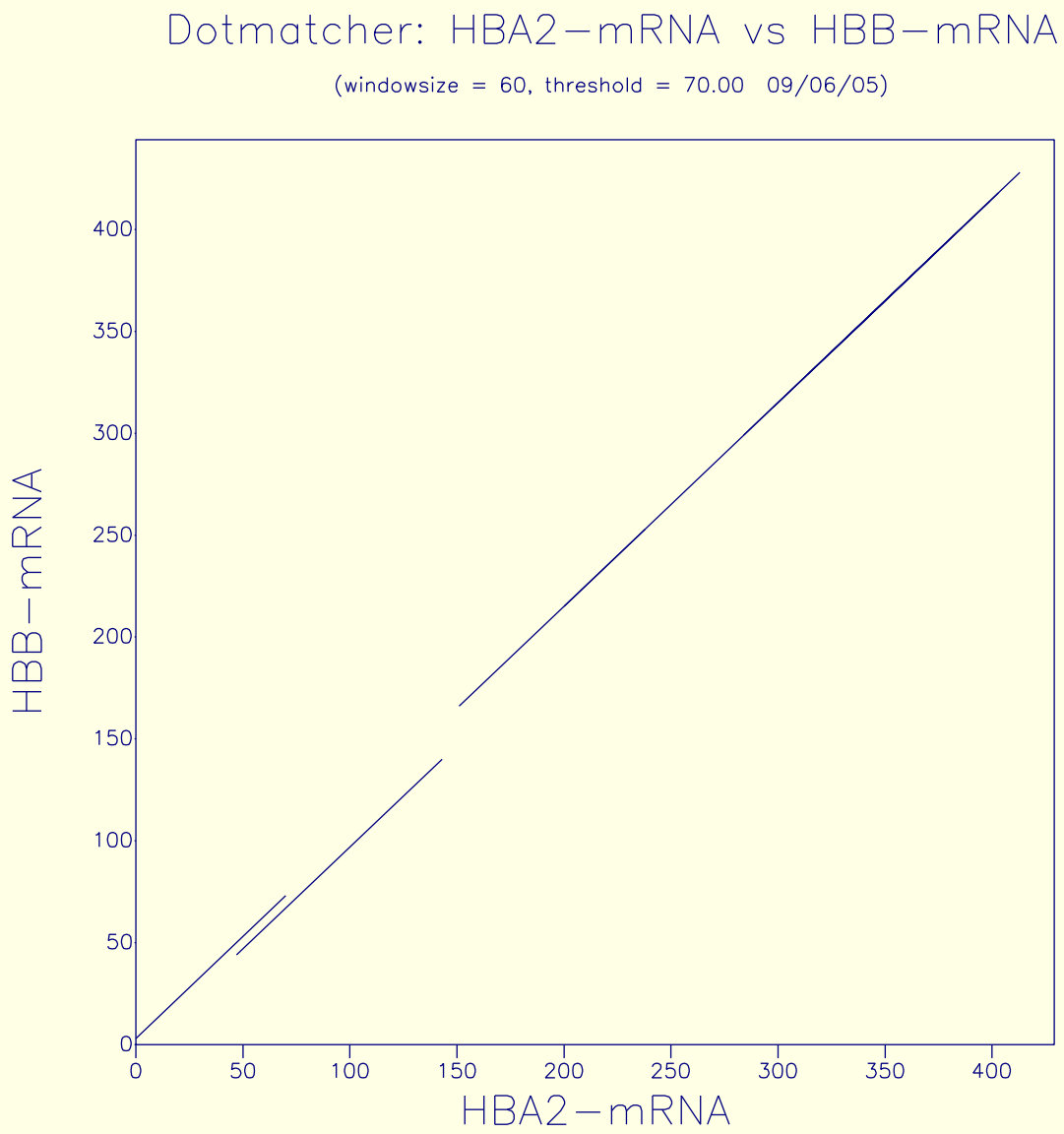
Taille fenêtre = 12

Dotplot sur des séquences réelles



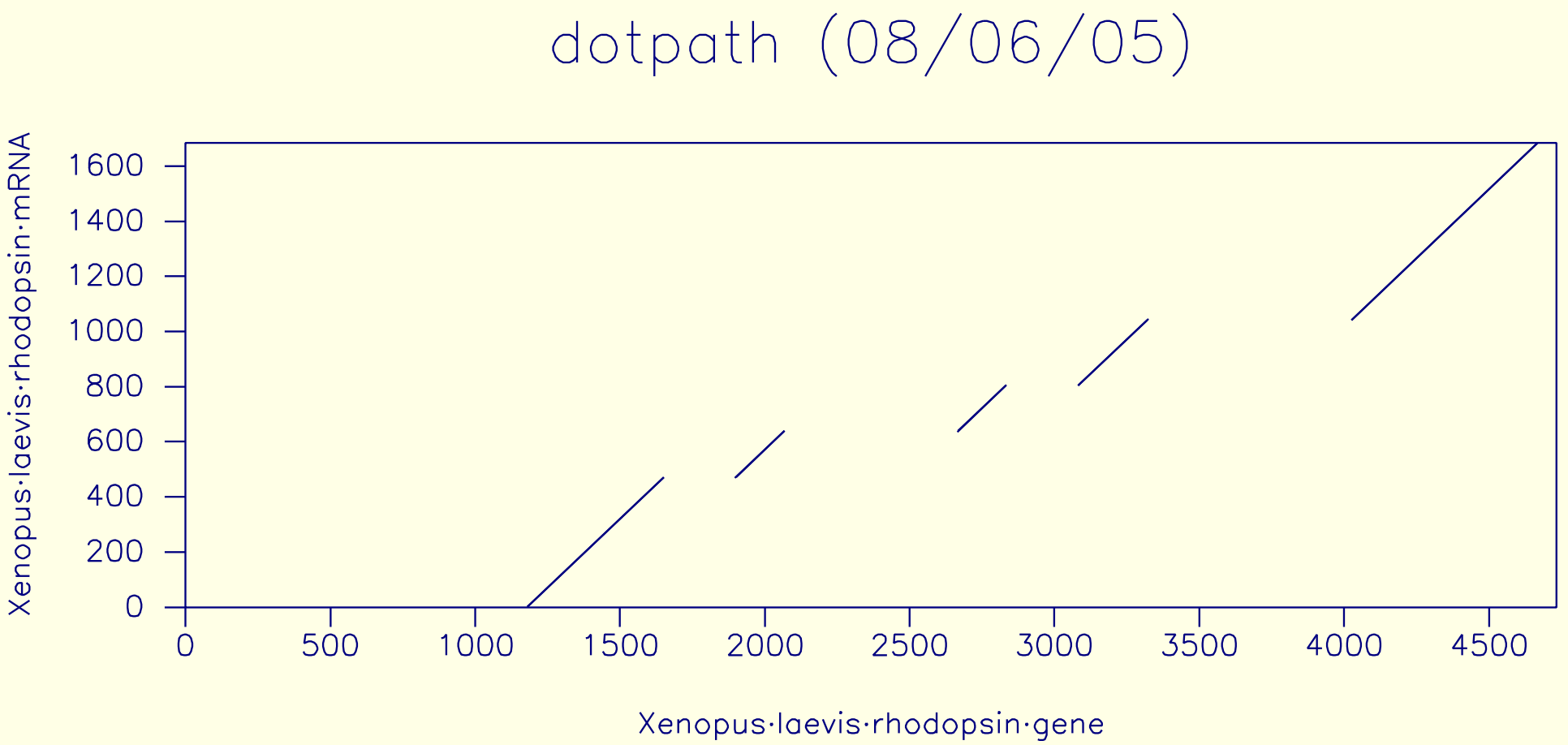
Taille fenêtre = 20

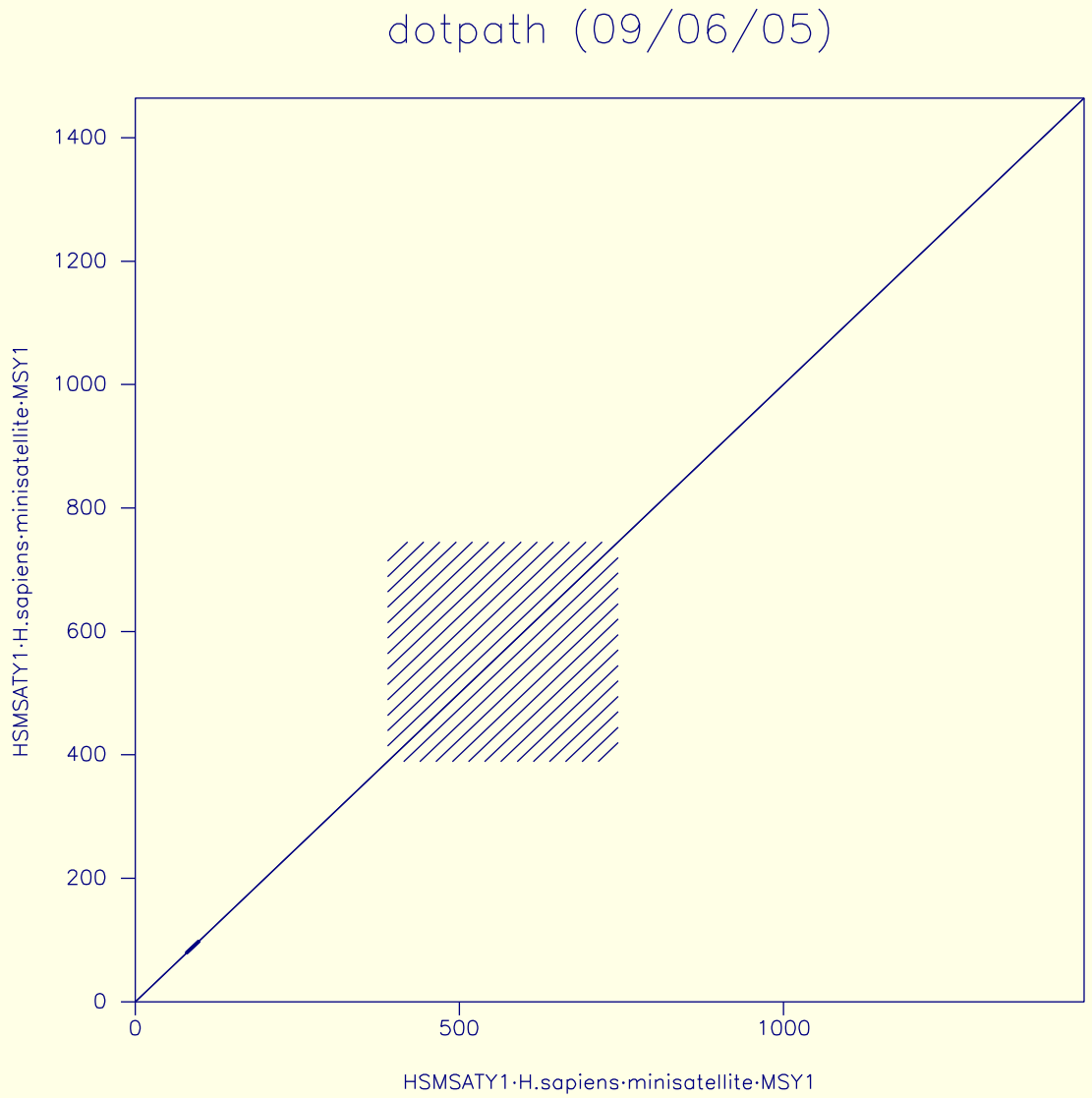
Repérer des similarités globales



Hémoglobine humaine chaînes $\alpha 2$ et β

Repérer des similarités locales





Minisatellite humain MSY1, taille de la fenêtre = 20

Avantage et inconvénients du dotplot

Avantages :

- → simple, visuel
- → très informatif
- → pas ou peu de perte d'information

Inconvénients :

- → identification ⇒ pas de méthode de détection automatique
- → interprétation ⇒ pas de mesure objective et d'évaluation de la pertinence de l'information

⇒ Besoin d'une mesure quantitative de similarité

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

$$\begin{array}{cccccccccccccc} R & D & I & S & L & V & - & - & - & K & N & A & G & I \\ | & & | & & | & | & & & & | & | & & | & | \\ R & N & I & - & L & V & S & D & A & K & N & V & G & I \end{array}$$

Qu'est-ce qu'un alignement ?

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

$$\begin{array}{ccccccccccccc} R & D & I & S & L & V & - & - & - & K & N & A & G & I \\ | & & | & & | & | & & & & | & | & & | & | \\ R & N & I & - & L & V & S & D & A & K & N & V & G & I \end{array}$$

- 3 événements mutationnels élémentaires = 3 opérations
  - ▷ substitution
  - ▷ insertion
  - ▷ délétion

  } $indels$

Qu'est-ce qu'un alignement ?

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

$$\begin{array}{ccccccccccccc} R & D & I & S & L & V & - & - & - & K & N & A & G & I \\ | & & | & & | & | & & & & | & | & & | & | \\ R & N & I & - & L & V & S & D & A & K & N & V & G & I \end{array}$$

- 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

Qu'est-ce qu'un alignement ?

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

$$\begin{array}{ccccccccccccc} R & D & I & S & L & V & - & - & - & K & N & A & G & I \\ | & & | & & | & | & & & & | & | & & | & | \\ R & N & I & - & L & V & S & D & A & K & N & V & G & I \end{array}$$

- 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

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

Définition formelle d'un alignement

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 formelle d'un alignement

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 :

Définition formelle d'un alignement

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 :

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

Définition formelle d'un alignement

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 :

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

$\rightarrow$ où chaque colonne met en correspondance deux symboles :

soit $\binom{s_i}{t_j}$ ou $\binom{s_i}{-}$ ou $\binom{-}{t_j}$

Définition formelle d'un alignement

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 :

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

$\rightarrow$ où chaque colonne met en correspondance deux symboles :

soit $\binom{s_i}{t_j}$ ou $\binom{s_i}{-}$ ou $\binom{-}{t_j}$

$\rightarrow$ où les colonnes doivent respecter l'ordre des séquences

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

Définition formelle d'un alignement

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 :

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

$\rightarrow$ où chaque colonne met en correspondance deux symboles :

soit $\binom{s_i}{t_j}$ ou $\binom{s_i}{-}$ ou $\binom{-}{t_j}$

$\rightarrow$ où les colonnes doivent respecter l'ordre des séquences

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

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

Ensemble des alignements

Séquences $s := I$, $t := L$

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

Ensemble des alignements

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

Ensemble des alignements

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

  $\rightarrow$ matrice s de score de similarité (PAM, BLOSUM, etc.)

  $\rightarrow$ $s(a, b)$ = score d'alignement des résidus a et b

- Identité/substitution : Scores positifs/négatifs

  $\rightarrow$ matrice s de score de similarité (PAM, BLOSUM, etc.)

  $\rightarrow$ $s(a, b)$ = score d'alignement des résidus a et b

- Insertion/délétion (indels) : Scores négatifs

  fonction de pénalité $\begin{cases} \text{élémentaire : unitaire pour un indel} \\ \text{complexe : affine, logarithmique} \end{cases}$

- Identité/substitution : Scores positifs/négatifs

  $\rightarrow$ matrice s de score de similarité (PAM, BLOSUM, etc.)

  $\rightarrow$ $s(a, b)$ = score d'alignement des résidus a et b

- Insertion/délétion (indels) : Scores négatifs

  fonction de pénalité $\begin{cases} \text{élémentaire : unitaire pour un indel} \\ \text{complexe : affine, logarithmique} \end{cases}$

- Score de l'alignement : somme des scores des événements élémentaires qui le composent

- Identité/substitution : Scores positifs/négatifs

  $\rightarrow$ matrice s de score de similarité (PAM, BLOSUM, etc.)

  $\rightarrow$ $s(a, b)$ = score d'alignement des résidus a et b

- Insertion/délétion (indels) : Scores négatifs

  fonction de pénalité $\begin{cases} \text{élémentaire : unitaire pour un indel} \\ \text{complexe : affine, logarithmique} \end{cases}$

- Score de l'alignement : somme des scores des événements élémentaires qui le composent

$\Rightarrow$ Maximiser le score

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

Distance d'édition	Distance d'édition généralisée
$Lev\begin{pmatrix} a \\ b \end{pmatrix} = 1$	$Edit\begin{pmatrix} a \\ b \end{pmatrix} = Sub(a,b) \geq 0$
$Lev\begin{pmatrix} a \\ - \end{pmatrix} = 1$	$Edit\begin{pmatrix} a \\ - \end{pmatrix} = Del(a) \geq 0$
$Lev\begin{pmatrix} - \\ b \end{pmatrix} = 1$	$Edit\begin{pmatrix} - \\ b \end{pmatrix} = Ins(b) \geq 0$
	Seule l'identité = 0

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

Distance d'édition	Distance d'édition généralisée
$Lev\begin{pmatrix} a \\ b \end{pmatrix} = 1$	$Edit\begin{pmatrix} a \\ b \end{pmatrix} = Sub(a,b) \geq 0$
$Lev\begin{pmatrix} a \\ - \end{pmatrix} = 1$	$Edit\begin{pmatrix} a \\ - \end{pmatrix} = Del(a) \geq 0$
$Lev\begin{pmatrix} - \\ b \end{pmatrix} = 1$	$Edit\begin{pmatrix} - \\ b \end{pmatrix} = Ins(b) \geq 0$

Seule l'identité = 0

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

$\Rightarrow$ Minimiser la distance

Les matrices de score

Les matrices nucléiques

- Peu de matrice
- Matrice unitaire, de transition/transversion, ...

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., convient pour des séquences avec un 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, convient pour la recherche de similarités locales, BLOSUM N : seuil de similarité

⇒ Explications détaillées au CM2

Fonction de gap (1/4)

- Fonction de gap = pénalité associée aux gaps
- Biologie : insertion d'un segment dans un gène
  $\Rightarrow$ les insertions/délétions arrivent en groupe
- gap : suite de l insertions ou de l délétions
- Mais pour ne pas donner lieu à des scénarios biologiquement invraisemblables, il faut les garder en nombre raisonnable :
  $\rightarrow$ 1 indel pour 20 a.a.

Fonction de gap (2/4)

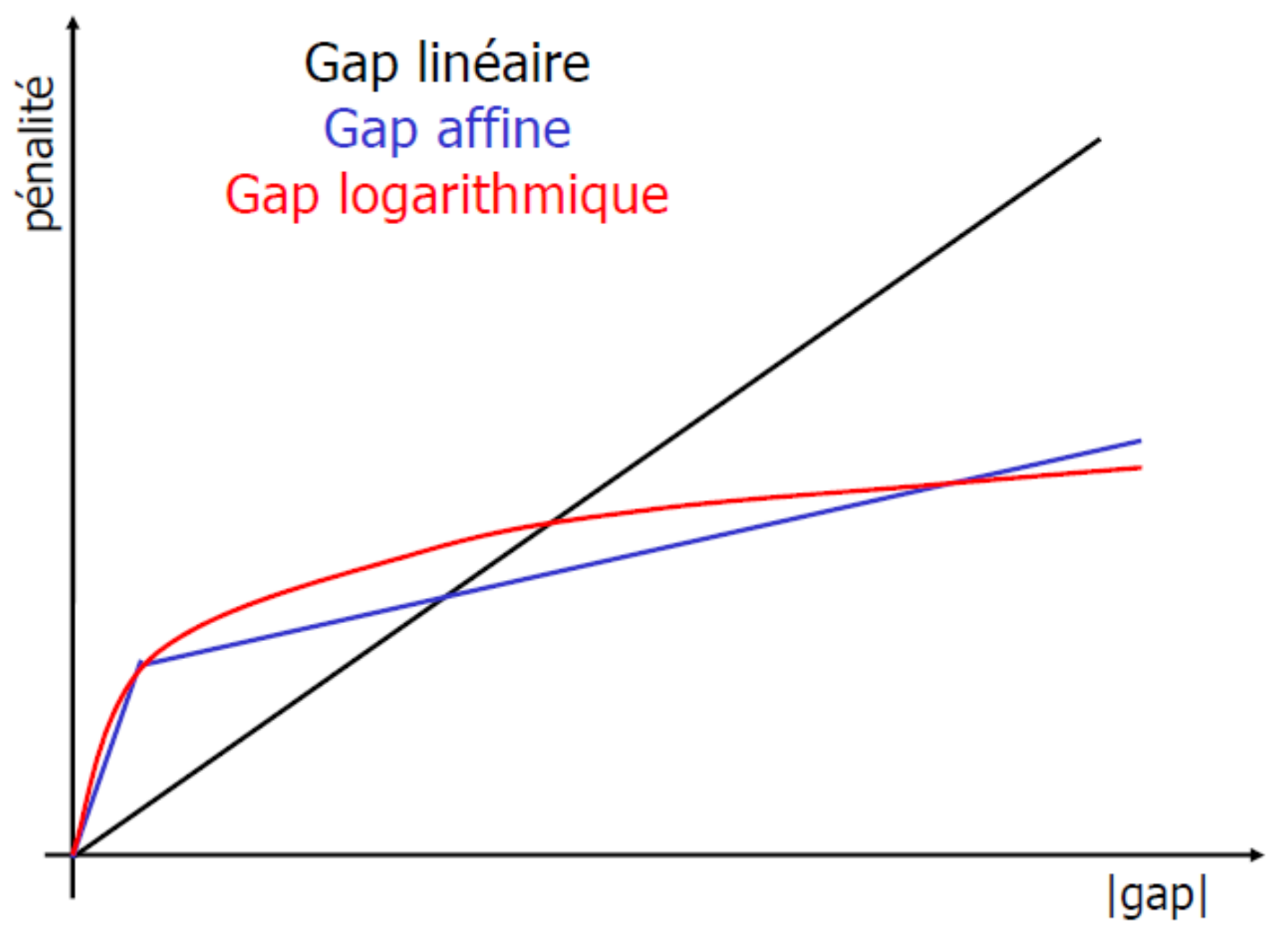
- Différentes fonctions de pénalités avec comme paramètres
  - o : pénalité d'ouverture de gap
  - e : pénalité d'extension de gap
  - l : longueur du gap

  → Fonction linéaire : $l \times e$
  $\Rightarrow$ besoin d'un pénalisation moindre que la fonction linéaire
  → Fonctions affines : $o + e \times (l - 1)$ où $o > l$
  → Fonctions logarithmiques : $o + e \times log(l)$
  → Fonctions qui pondère e par la distance évolutive ou la nature des résidus
  → ...

- Fonctions de pénalité les plus réalistes = fonctions affines.

- Fonctions de gaps différentes $\Rightarrow$ alignements différents

Fonction de gap (3/4)



Fonction de gap (4/4)

- Exemple d'alignements possibles avec fonction de gap affine

```
fonction affine : o + e x longueur_extension
o      0  1   1   1
e      0  0  0.1  1
-----------------------------------
Al1    7  5  4.9  4    ATGCGggACaTG
                       AgGCG--cC-TG   (7 id., 1 gap d'1 pb, 1 gap de 2 pb)
Al2    7  5  4.9  4    ATGCGGgaCaTG
                       AgGCGc--C-TG   (7 id., 1 gap d'1 pb, 1 gap de 2 pb)
Al3    7  6  5.8  4    ATGCGggaCATG
                       AgGCG---CcTG   (7 id., 1 gap de 3 pb)
Al4    7  4   4   4    ATGCGgGaCaTG
                       AgGCG-c-C-TG   (7 id., 3 gaps de 1 pb)
```

- Importance des pénalités de gaps (voir TP)

  Pénalité différente $\Rightarrow$ alignement différent

- Remarque : le score le plus élevé n'est pas nécessairement celui du meilleur alignement

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

Exemple

- Aligner les séquences : $ACGGCTATC$ et $ACTGTAATG$
- Scores : indel $= -1$, mismatch -2, match $= 2$

Exemple

- Aligner les séquences : $ACGGCTATC$ et $ACTGTAATG$
- Scores : indel $= -1$, mismatch -2, match $= 2$

- Alignement optimal complet, dernière opération : 3 possibilités

Exemple

- Aligner les séquences : $ACGGCTATC$ et $ACTGTAATG$
- Scores : indel $= -1$, mismatch -2, match $= 2$

- Alignement optimal complet, dernière opération : 3 possibilités

substitution		délétion		insertion	
ACGGCTATC		ACGGCTAT C		ACGGCTATC-	
? ? ?	-2	? ? ?	-1	? ? ?	-1
ACTGTAATG		ACTGTAATG-		ACTGTAAT G	

Exemple

- Aligner les séquences : $ACGGCTATC$ et $ACTGTAATG$
- Scores : indel $= -1$, mismatch -2, match $= 2$

- Alignement optimal complet, dernière opération : 3 possibilités

substitution		délétion		insertion	
ACGGCTATC		ACGGCTAT C		ACGGCTATC-	
? ? ?	-2	? ? ?	-1	? ? ?	-1
ACTGTAATG		ACTGTAATG-		ACTGTAAT G	

Problème sur des séquences plus courtes

Exemple

- Aligner les séquences : $ACGGCTATC$ et $ACTGTAATG$
- Scores : indel $= -1$, mismatch -2, match $= 2$

- Alignement optimal complet, dernière opération : 3 possibilités

substitution		délétion		insertion	
ACGGCTATC		ACGGCTAT C		ACGGCTATC-	
? ? ?	-2	? ? ?	-1	? ? ?	-1
ACTGTAATG		ACTGTAATG-		ACTGTAAT G	

Problème sur des séquences plus courtes

$\Rightarrow$ 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 / résolution sous-problèmes

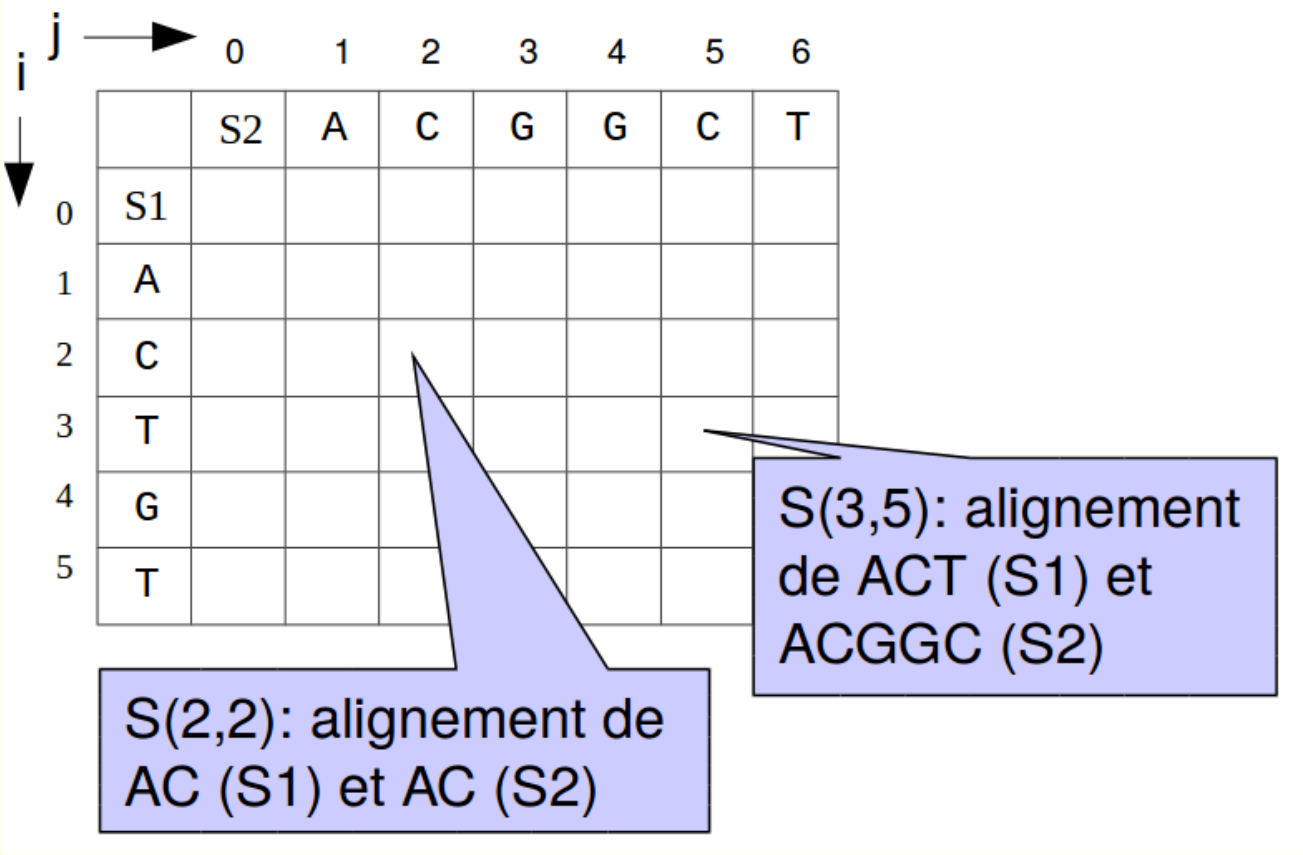
=

calculs scores d'alignements entre préfixes

⇒ combinaison des sous-alignements précédents

Alignement global [Needleman & Wunsch, 1970]

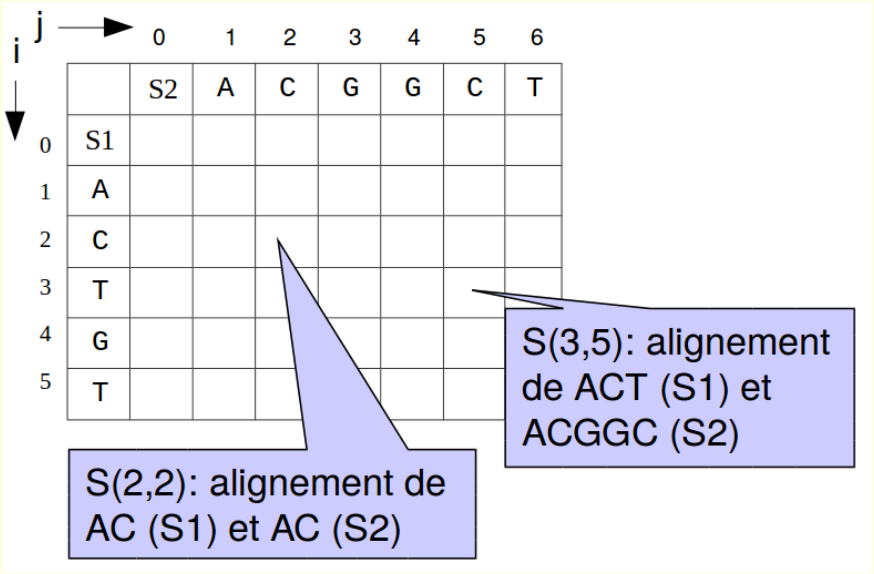
⇒ Application du principe de la programmation dynamique = combinaison des sous-alignements précédents



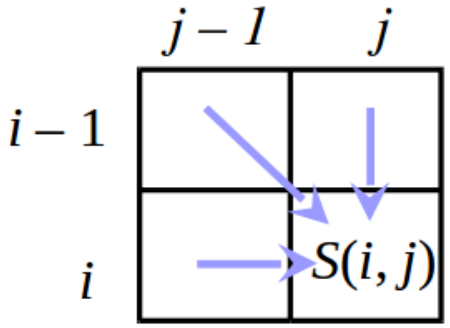
Case (i, j) : score alignement entre les i premières bases de ACGGCT et les j premières bases de ACTGT

Alignement global [Needleman & Wunsch, 1970]

⇒ Application du principe de la programmation dynamique = combinaison des sous-alignements précédents



S(i,j) = coût optimal pour les i (j) premières lettres de S1 (S2)



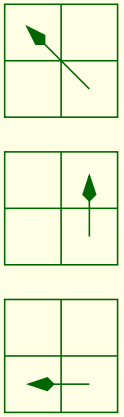
Coût aln		Coût aln	Coût
ACGGC ACT	=	ACGG AC	+ subs(**C**,**T**)
ACGGC ACT	=	ACGGC AC	+ indel(-, **T**)
ACGGC ACT	=	ACGG ACT	+ indel(**C**, -)

- 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

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



Un exemple pas à pas

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

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

Un exemple pas à pas

		A	C	G	G	C	T	A	T	C
	0									
A										
C										
T										
G										
T										
A										
A										
T										
G										

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

Un exemple pas à pas

		A	C	G	G	C	T	A	T	C
	0	-1								
A										
C										
T										
G										
T										
A										
A										
T										
G										

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

Un exemple pas à pas

		A	C	G	G	C	T	A	T	C
	0	-1	-2							
A										
C										
T										
G										
T										
A										
A										
T										
G										

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

Un exemple pas à pas

		A	C	G	G	C	T	A	T	C
	0	-1	-2	-3						
A										
C										
T										
G										
T										
A										
A										
T										
G										

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

Un exemple pas à pas

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

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

Un exemple pas à pas

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

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

Un exemple pas à pas

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

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

Un exemple pas à pas

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

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

Un exemple pas à pas

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

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

Un exemple pas à pas

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

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

Un exemple pas à pas

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

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

Un exemple pas à pas

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

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

Un exemple pas à pas

		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									
T	-3									
G	-4									
T	-5									
A	-6									
A	-7									
T	-8									
G	-9									

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

Un exemple pas à pas

		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								
T	-3									
G	-4									
T	-5									
A	-6									
A	-7									
T	-8									
G	-9									

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

Un exemple pas à pas

		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							
T	-3									
G	-4									
T	-5									
A	-6									
A	-7									
T	-8									
G	-9									

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

Un exemple pas à pas

		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						
T	-3									
G	-4									
T	-5									
A	-6									
A	-7									
T	-8									
G	-9									

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

Un exemple pas à pas

		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									
G	-4									
T	-5									
A	-6									
A	-7									
T	-8									
G	-9									

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

Un exemple pas à pas

		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								
G	-4									
T	-5									
A	-6									
A	-7									
T	-8									
G	-9									

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

Un exemple pas à pas

		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							
G	-4									
T	-5									
A	-6									
A	-7									
T	-8									
G	-9									

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

Un exemple pas à pas

		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									
T	-5									
A	-6									
A	-7									
T	-8									
G	-9									

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

Un exemple pas à pas

		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 ; $g = -1$

Un exemple pas à pas

		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

- Score du meilleur alignement global $\Rightarrow$ valeur de la case en bas à droite dans 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 $+$ la mutation correspondante $=$ 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.

Backtracking : exemple

		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**

Backtracking : exemple

		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**

G

C

Backtracking : exemple

		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**

$$\begin{array}{cc} T & G \\ | & \\ T & C \end{array}$$

Backtracking : exemple

		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**

$$\begin{array}{ccc} A & T & G \\ & | & \\ \_ & T & C \end{array}$$

Backtracking : exemple

		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**

$$\begin{array}{cccc} A & A & T & G \\ | & & | & \\ A & \_ & T & C \end{array}$$

Backtracking : exemple

		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**

$$\begin{array}{ccccc} T & A & A & T & G \\ | & | & & | & \\ T & A & \_ & T & C \end{array}$$

Backtracking : exemple

		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**

$$
\begin{array}{cccccc}
\_ & T & A & A & T & G \\
 & | & | & & | & \\
C & T & A & \_ & T & C
\end{array}
$$

Backtracking : exemple

		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**

$$
\begin{array}{cccccccc}
G & \_ & T & A & A & T & G \\
| & & | & | & & | & \\
G & C & T & A & \_ & T & C
\end{array}
$$

Backtracking : exemple

		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**

$$\begin{array}{cccccccc} T & G & \_ & T & A & A & T & G \\ & | & & | & | & & | & \\ G & G & C & T & A & \_ & T & C \end{array}$$

Backtracking : exemple

		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**

$$
\begin{array}{cccccccccc}
C & T & G & \_ & T & A & A & T & G & \\
 & | & & | & & | & | & & | & \\
C & G & G & C & T & A & \_ & T & C &
\end{array}
$$

Backtracking : exemple

		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**

$$
\begin{array}{ccccccccccc}
A & C & T & G & \_ & T & A & A & T & G \\
| & | & & | & & | & | & & | & \\
A & C & G & G & C & T & A & \_ & T & C
\end{array}
$$

		G	G	C	T	G	A	C
G								
A								
T								
C								

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* $a = b$

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

- 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.

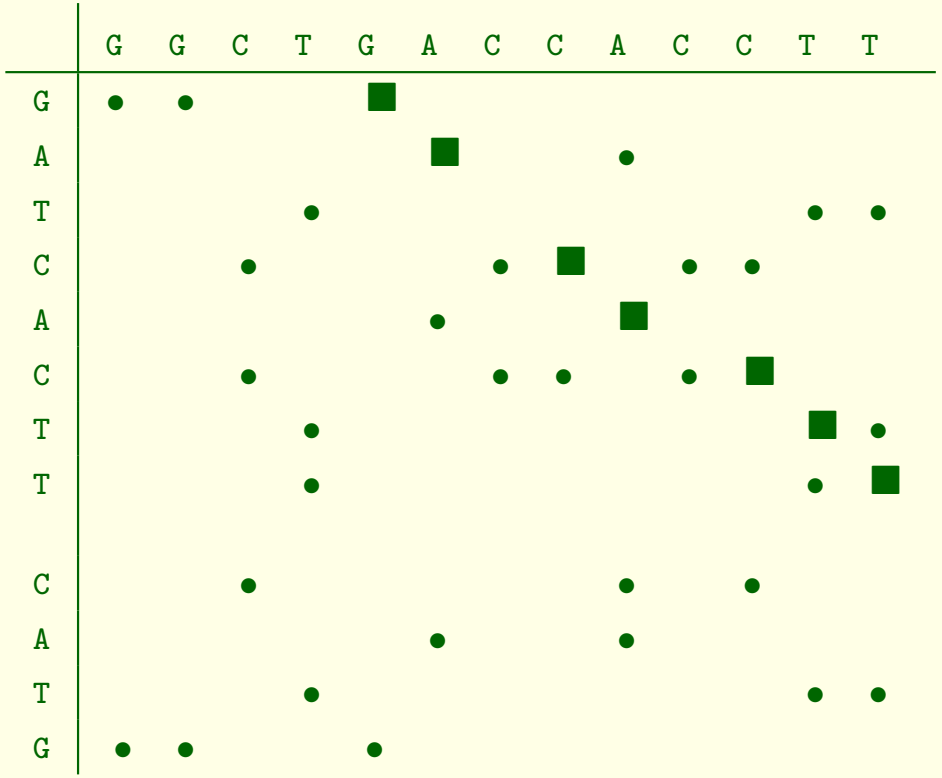
Alignement local : un exemple

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.

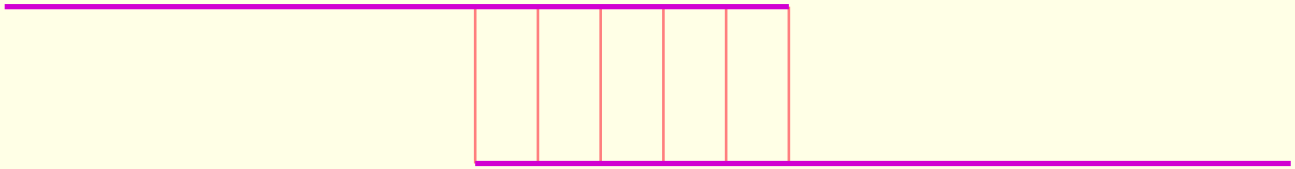
- s : une matrice de score ;
- g : pénalité associée à un indel ;
- Initialisation :
$$\begin{cases} M(0,0) = 0 \\ M(0,j) = 0 & \longleftarrow \\ M(i,0) = 0 & \longleftarrow \end{cases}$$
- Remplissage :
$$M(i,j) = \max \begin{cases} M(i-1,j-1) + s(x_i, y_j) & \text{Sub. ou App. exact} \\ M(i-1,j) + g & \text{Délétion} \\ M(i,j-1) + g & \text{Insertion} \\ 0 & \longleftarrow \end{cases}$$
- Score du meilleur alignement local $\Rightarrow$ valeur max dans la matrice.

Voir feuille de TD

Alignement semi-global : le principe

- 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



Alignement semi-global

```
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 - - - - - - -
```

$\Rightarrow$ L'alignement global préfère le 1^{er} alignement.

- Initialisation $\Rightarrow$ idem à l'alignement local : $\begin{cases} M(0,0) = 0 \\ M(0,j) = 0 & \longleftarrow \\ M(i,0) = 0 & \longleftarrow \end{cases}$

- Remplissage $\Rightarrow$ idem à l'alignement global

- Score du meilleur alignement semi-local $\Rightarrow$ valeur max dans la dernière ligne et la dernière colonne de la matrice.

Alignement avec fonction affine de gaps (1/2)

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

- 3 matrices
  - $\rightarrow$ Match/mismatch (M) : score max d'un alignement qui finit par une substitution
  - $\rightarrow$ Délétion (D) : score max d'un alignement qui finit par une délétion
  - $\rightarrow$ Insertion (I) : score max d'un alignement qui finit par une insertion

- Initialisation : $\begin{cases} M(0,0) = D(0,0) = I(0,0) = 0 \\ M(0,j) = D(0,j) = I(0,j) = c(j) \\ M(i,0) = D(i,0) = I(i,0) = c(i) \end{cases}$

- 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 :

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

Exercice 3

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 $O(|T| \times |P|)$.

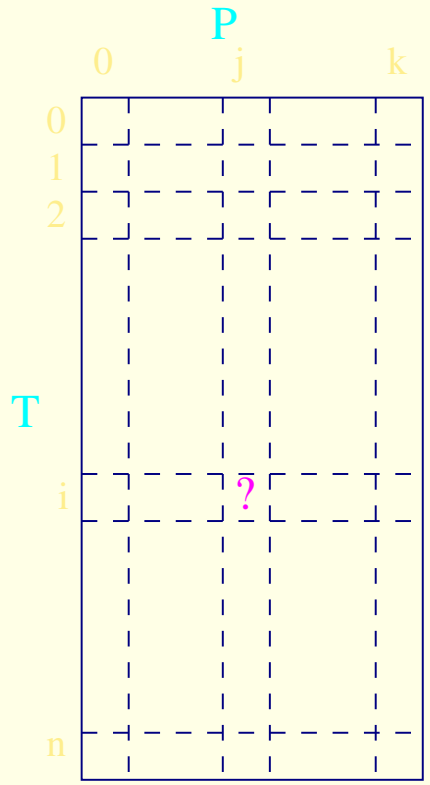
- Algorithme de [Myers et Miller 89] et [Fischetti et al. 92]

- 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 :

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

$\rightarrow$ Récurrence générale : 2 passes par ligne.

- 1^{re} passe :

$$\mathcal{M}(i,1) = \min \begin{cases} \mathcal{M}(i-1,0) + Sub(T[i],P[1]) & \text{Sub}(T[i],P[1]) \\ \mathcal{M}(i-1,k) + Sub(T[i],P[1]) & \text{WSub}(T[i],P[1]) \\ \mathcal{M}(i-1,1) + Del & \text{Délétion de } T[i] \\ \mathcal{M}(i,0) + Ins & \text{Insertion de } P[1] \end{cases}$$

$$\mathcal{M}(i,j) = \min \begin{cases} \mathcal{M}(i-1,j-1) + Sub(T[i],P[j]) & \text{Sub}(T[i],P[j]) \\ \mathcal{M}(i-1,j) + Del & \text{Délétion de } T[i] \\ \mathcal{M}(i,j-1) + Ins & \text{Insertion de } P[j] \end{cases}$$

- 1^{re} passe :

$$\mathcal{M}(i,1) = \min \begin{cases} \mathcal{M}(i-1,0) + Sub(T[i],P[1]) & \text{Sub}(T[i],P[1]) \\ \mathcal{M}(i-1,k) + Sub(T[i],P[1]) & \text{WSub}(T[i],P[1]) \\ \mathcal{M}(i-1,1) + Del & \text{Délétion de } T[i] \\ \mathcal{M}(i,0) + Ins & \text{Insertion de } P[1] \end{cases}$$

$$\mathcal{M}(i,j) = \min \begin{cases} \mathcal{M}(i-1,j-1) + Sub(T[i],P[j]) & \text{Sub}(T[i],P[j]) \\ \mathcal{M}(i-1,j) + Del & \text{Délétion de } T[i] \\ \mathcal{M}(i,j-1) + Ins & \text{Insertion de } P[j] \end{cases}$$

- 2^{e} passe :

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

$$\mathcal{M}(i,j) = \min \begin{cases} \mathcal{M}(i,j) & \text{Valeur précédente} \\ \mathcal{M}(i,j-1) + Ins & \text{Insertion de } P[j] \end{cases}$$

Voir feuille de TD

Significativité des scores obtenus

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 dans une banque de séquences

Approche empirique : Tests de permutation (TP logiciel PRSS)

1. Réarranger de manière aléatoire les lettres d'une des séquences
2. Produire l'alignement des séquences permutées
3. Conserver son score
4. Répéter les étapes 1 à 3 un très grand nombre de fois

Exemples avec 200 permutations :

- ILV1_ILVB.prss : score 2644 et scores aléatoires $[31 - 50]$
- DCP1_ILVB.prss : score 129 et scores aléatoires $[27 - 53]$
- ILVB_EGR1.prss : score 86 et score aléatoires $[60 - 127]$

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

- décrit le bruit aléatoire qui existe lorsqu'on aligne des séquences
- croit de manière proportionnelle en fonction de n et m
- décroit de manière exponentielle en fonction du score S
- plus la E-value est proche de 0, plus la similarité est significative

- Alignement par paire global [Needleman Wunsch, 1970]
  - → Le meilleur alignement sur toute leur longueur
  - → Pour des séq très similaires et$\sim$ de émême longueur
    - ⇒ Needle (EMBOSS) : algorithme de N-W
    - ⇒ Stretcher (EMBOSS) : version rapide (longues séq)

- Alignement par paire local [Smith Waterman, 1981]
  - → Les régions les plus similaires dans les 2 séquences
  - → Trouver des sous-séq ayant des relations biologiques
    - ⇒ Water (EMBOSS) : version rapide de l'algo de S-W
    - ⇒ Matcher (EMBOSS) : implémentation d'un algo rigoureux basé sur le logiciel Lalign
      (Lalign : recherche des duplications internes en calculant les alignements locaux sans intersection)

- Comparaison des logiciels d'alignement par paire

  → Stretcher et Matcher utilisent beaucoup moins d'espace mémoire

  → Needle et Water guarantissent une solution optimale

  → Matcher renvoie plusieurs alignements locaux trouvées alors que Water renvoie l'alignement local optimal uniquement

- Méthodes graphique d'alignement par paire (dotplot)

  → dotpath (EMBOSS)

  → dotmatcher (EMBOSS)

- Introduction
- Dotplot
- Alignement : définitions et scores
- Algorithme d'alignement (programmation dynamique)
- Autres types d'alignements
- Significativité
- Les logiciels
- 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. $=$ $\underline{\sim 3 \text{ heures}}$

  $\Rightarrow$ Nécessité d'algorithmes plus rapides

- Algorithmes heuristiques, les plus connus : BLAST et FASTA