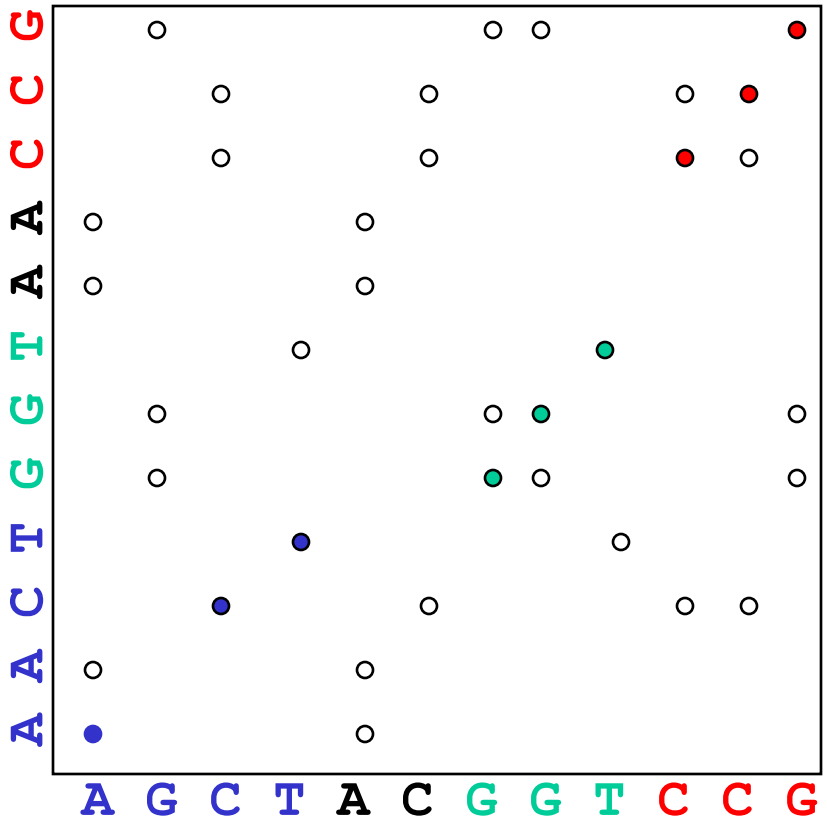
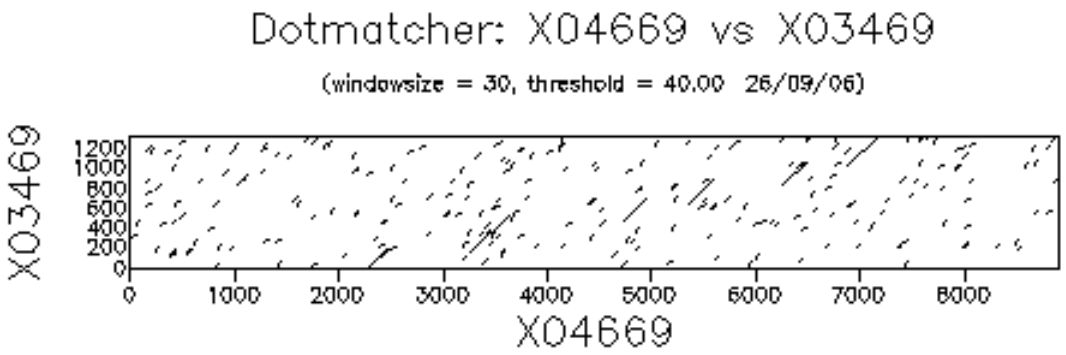
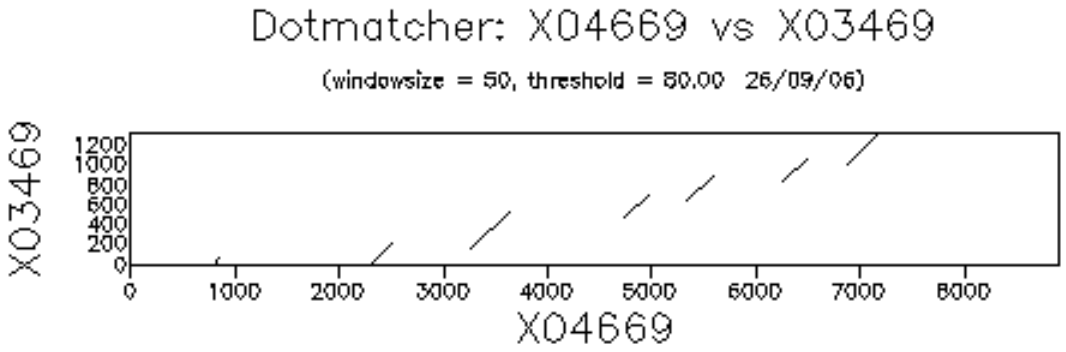
Comparaison de deux séquences

Matrice de points (Dotplot)



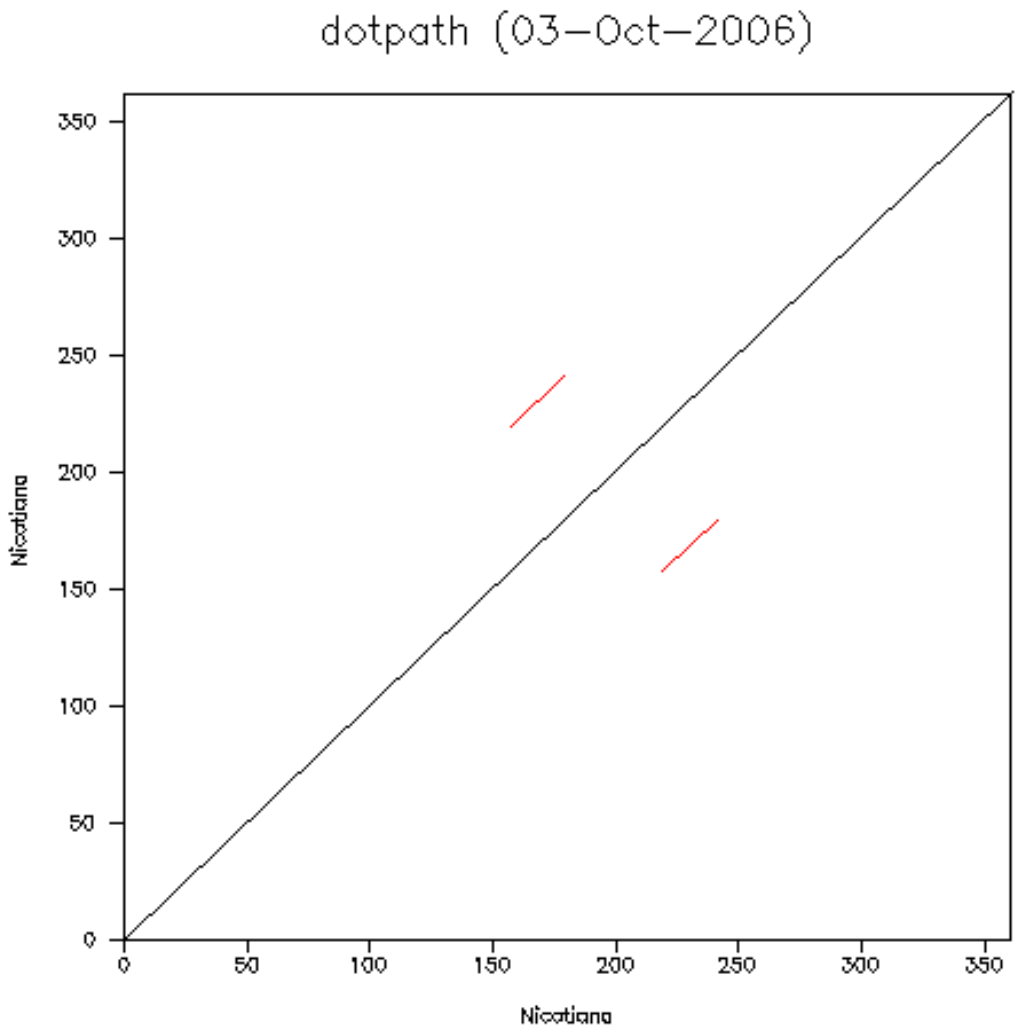
Exemple d'utilisation d'un dotplot : mise en évidence des exons



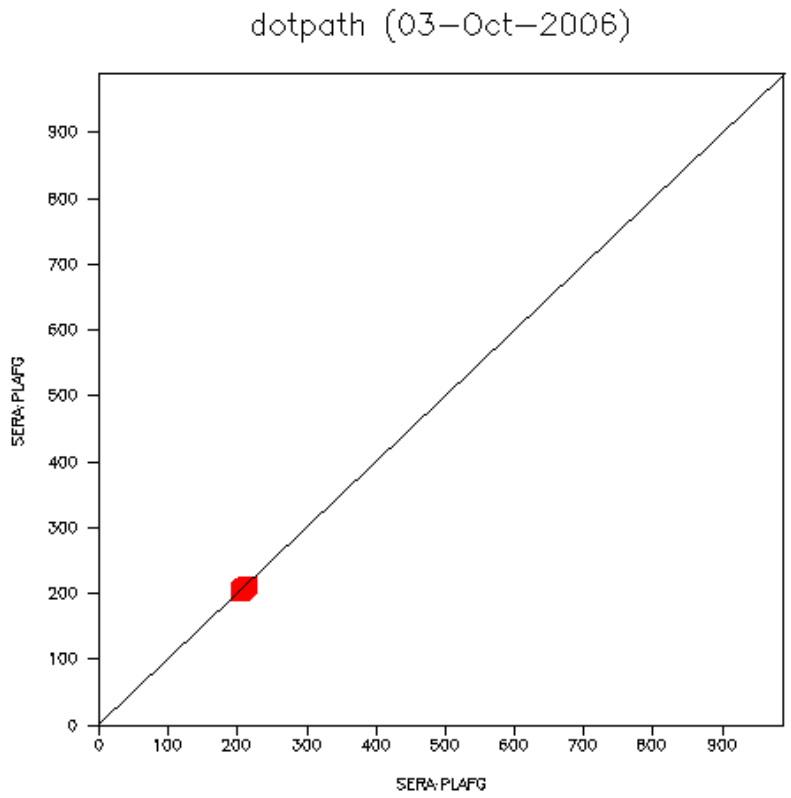


Effet des paramètres pour filtrer le bruit de fond

Exemple d'utilisation d'un dotplot : mise en évidence de répétitions



Exemple d'utilisation d'un dotplot : mise en évidence de « structure »



>SERA_PLAFG (P13823): Plasmodium falciparum serine-repeat antigen protein precursor.
MKSYISLFFILCVIFNKNVIKCTGESQTGNTGGGQAGNTVGDQAGSTGGSPQG
STGASQPGSSEPSNPVSSGHSVSTVSVSQTSTSSEKQDTIQVKSALLKDYMGL
KVTGPCNENFIMFLVPHIYIDVDTEDTNIELRTTLKETNNAISFESNSGSLEK
KKYVLPSNGTTGEQGSSTGTVRGDTEPISDSSSSSSSSSSSSSSSSSSSSSS
SSSSSSSSSSSSESLPANGPDSPTVKPPRNLQNICETGKNFKLVVYIKENTLI
IKWKVYGETKDTTENNKVDVRKYLINEKETPFTSILIHAYKEHNGTNLIESKN
YALGSDIPEKCDTLASNCFLSGNFNIEKCFQCALLVEKENKNDVCYKYLSEDI
VSNFKEIKAETEDDDEDDYTEYKLTESIDNILVKMFKTNENNDKSELIKLEEV
DDSLKLELMNYCSLLKDVDTTGTLDNYGMGNEMDIFNNLKRLLIYHSEENINT
LKNKFRNAAVCLKNVDDWIVNKRGLVLPELNYDLEYFNEHLYNDKNSPEDKDN
KGKGVVHVDTTLEKEDTLSYDNSDNMFCNKEYCNRLKDENNCISNLQVEDQGN
CDTSWIFASKYHLETIRCMKGYEPTKISALYVANCYKGEHKDRCDEGSSPMEF
LQIIEDYGFLPAESNYPYNYVKVGEQCPKVEDHWMNLWDNGKILHNKNEPNSL
DGKGYTAYESERFHDNMDAFVKIIKTEVMNKGSVIAYIKAENVMGYEFSGKKV
QNLCGDDTADHAVNIVGYGNYVNSEGEKKSYWIVRNSWGPYWGDEGYFKVDMY
GPTHCHFNFIHSVVIFNVDLPMNNKTTKKESKIYDYYLKASPEFYHNLYFKNF
NVGKKNLFSEKEDNENNKKLGNNYIIFGQDTAGSGQSGKESNTALESAGTSNE
VSERVHVYHILKHIKDGKIRMGMRKYIDTQDVNKKHSCTRSYAFNPENYEKCV
NLCNVNWKTCEEKTSPGLCLSKLDTNNECYFCYV

Alignement de deux séquences

La matrice de point est un bon outil visuel permettant de détecter les régions conservées entre deux séquences mais elle est insuffisante car on ne peut pas quantifier les similitudes observées

➡ Calcul d'un score

Alignement déduit du premier dotplot:

```
AACT--GGTAACCG
AGCTACGGT--CCG
```

Le score de l'alignement doit prendre en compte toutes les positions alignées : identités, substitutions et indels. Chacun de ces évènements va recevoir un poids, appelé score élémentaire $s_{e.}$. Le score de l'alignement correspondra à la somme des scores élémentaires correspondant aux positions alignées.

$$S = \sum_{i=1}^{l} Se(i)$$

Où l est le nombre de positions alignées

exemple: l = 14

s_e identité = +2
s_e substitution = -1
s_e indels = -2

➡ S = 9

Algorithme de programmation dynamique

Etant donné un système de score, garantit l'obtention de l'alignement optimal

Hypothèse : l'évolution est parcimonieuse
Signification: pour trouver l'alignement optimal, l'algorithme va rechercher le chemin permettant de passer d'une séquence à l'autre avec le minimum de changements

Deux types de score en fonction des algorithmes :

- score d'homologie: la valeur du score diminue avec le nombre de différences observées entre les deux séquences

- score de distance: la valeur du score augmente avec le nombre de différences observées entre les deux séquences

Exemples de systèmes de scores

	Score d'homologie	Score de distance
identité	+1	0
mismatch	-1	+1
indel	-2	+2

Algorithme de programmation dynamique

Trois types d'algorithmes d'alignement de deux séquences:

• alignement global (proposé en premier par Needleman and Wunsch). Les séquences vont être alignées sur toutes leurs longueurs (du premier au dernier résidus). Utilisé quand les séquences ont à peu près la même longueur

• alignement semi-global (pas de pénalités des gaps aux extrémités). Utilisé quand une séquence est plus courte que l'autre ou quand on recherche des chevauchements aux extrémités.

• alignement local (connu comme l'algorithme de Smith and Waterman). L'algorithme recherche les deux sous-régions les plus conservées entre les deux séquences. Seulement ces deux régions seront alignées.

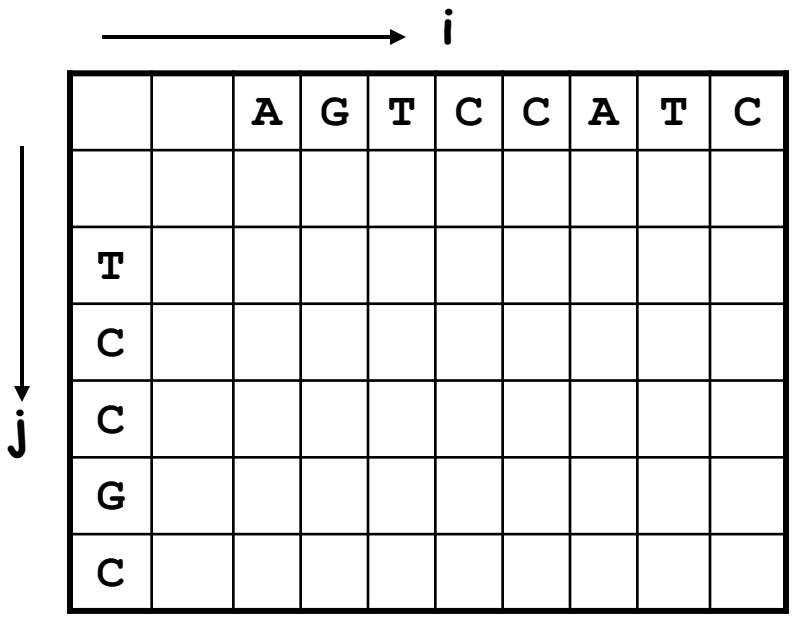
Algorithme de programmation dynamique

Comment ça marche ?

Prenons comme exemple deux séquences X et Y de longueur M et N :

X = AGTCCATC M=8
Y = TCCGC N=5

Matrice de programmation dynamique :



Le score optimal sera calculé récursivement. Le score calculé pour la cellule (i,j) correspondra au meilleur alignement des résidus $x_1 \ldots\ldots x_i$ avec les résidus $y_1 \ldots\ldots y_j$

Algorithme de programmation dynamique

Comment calcule-t-on le score d'une cellule ?

Il y a seulement trois façons d'aligner x_i avec y_j :

- les deux résidus s'alignent (identité ou substitution)
- le résidu x_i est aligné avec un gap (insertion dans la séquence X)
- le résidu y_j est aligné avec un gap (insertion dans la séquence Y)

Cela correspond à trois chemins différents pour atteindre la cellule (i,j)

- on atteint la cellule par la diagonale en venant de la cellule (i-1,j-1)
- on atteint la cellule par l'horizontale en venant de la cellule (i-1,j)
- on atteint la cellule par la verticale en venant de la cellule (i, j-1)

L'algorithme doit choisir entre ces trois chemins. S'il utilise un score d'homologie, il choisira le chemin qui maximise la valeur du score s(i,j). S'il utilise un score de distance, il choisira le chemin qui minimise la valeur du score s(i,j).

Algorithme de programmation dynamique

Score d'homologie:

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

Score de distance:

$$s(i,j) = \min \begin{cases} s(i-1, j-1) + w(x_i, y_j) \\ s(i-1, j) + \delta \\ s(i, j-1) + \delta \end{cases}$$

Où :

- $w(x_i, y_j)$ est la valeur dans le système de score correspondant soit à l'identité soit à la substitution (mismatch)
- δ est le poids de l'insertion/délétion (indel)

Algorithme de programmation dynamique

Une fois la matrice remplie, le score de l'alignement optimal est le dernier calculé s(M,N). Mais on ne connaît pas encore l'alignement proprement dit. Il va être construit par une procédure de « retour en arrière » récursive. En partant de la dernière cellule (M,N), on détermine le chemin utilisé pour l'atteindre et on le traduit en terme d'alignement. On continue le processus, une cellule du chemin optimal à la fois, jusqu'à atteindre la première cellule (1,1). A ce point, l'alignement optimal est complètement reconstruit.

Alignement local:

Deux différences majeures:

- l'alignement peut commencer à n'importe quelles positions, pas forcément les premières
- l'alignement peut se terminer à n'importe quelles positions, pas forcément les dernières

L'algorithme va utiliser un score d'homologie et seule l'identité recevra un poids positif (score élémentaire).
Quand la valeur du score d'une cellule devient négatif, il est remplacé par zéro. Il vaut mieux recommencer un nouvel alignement que de le prolonger. Donc une cellule contenant un zéro indique le début d'un alignement.
Quand on reconstruit l'alignement par la procédure de « retour en arrière », au lieu de partir de la dernière cellule, on choisira celle qui a le score le plus élevé.

Alphabet étendu pour les nucléotides

- Problème de séquençage
- Polymorphisme

L'alphabet étendu permet de modéliser l'incertitude sur une séquence : le nucléotide à une position n'est pas clairement identifié ou peut varier.

Symbol	Meaning	Nucleic Acid
A	A	Adenine
C	C	Cytosine
G	G	Guanine
T	T	Thymine
U	U	Uracil
M	A or C	
R	A or G	purine
W	A or T	
S	C or G	
Y	C or T	pyrimidine
K	G or T	
V	A or C or G	not T
H	A or C or T	not G
D	A or G or T	not C
B	C or G or T	not A
X	G or A or T or C	any
N	G or A or T or C	any
.	G or A or T or C	any

Problème :
un mismatch entre *A* et *C* n'a pas le même coût qu'un mismatch entre *A* et *M* !

Système de score : matrices de substitution

	Score d'homologie	Score de distance
identité	+1	0
mismatch	-1	+1
indel	-2	+2

	A	T	C	G	-
A	+1	-1	-1	-1	-2
T	-1	+1	-1	-1	-2
C	-1	-1	+1	-1	-2
G	-1	-1	-1	+1	-2
-	-2	-2	-2	-2	

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

Les matrices de substitution permettent de spécifier le coût/score de chaque substitution possible (A avec C, A avec T, …) de manière indépendante

Matrice pour les nucléotides (alphabet étendu)

NUC4.4 pour BLAST ou EDNAFULL pour EMBOSS

```
# This matrix was created by Todd Lowe   12/10/92
#
# Uses ambiguous nucleotide codes, probabilities rounded to
#  nearest integer
#
# Lowest score = -4, Highest score = 5
#
```

	A	**T**	**G**	**C**	S	W	R	Y	K	M	B	V	H	D	N
A	5	-4	-4	-4	-4	1	1	-4	-4	1	-4	-1	-1	-1	-2
T	-4	5	-4	-4	-4	1	-4	1	1	-4	-1	-4	-1	-1	-2
G	-4	-4	5	-4	1	-4	1	-4	1	-4	-				
C	-4	-4	-4	5	1	-4	-4	1	-4	1	-				
S	-4	-4	1	1	-1	-4	-2	-2	-2	-2	-				
W	1	1	-4	-4	-4	-1	-2	-2	-2	-2	-				
R	1	-4	1	-4	-2	-2	-1	-4	-2	-2	-				
Y	-4	1	-4	1	-2	-2	-4	-1	-2	-2	-				
K	-4	1	1	-4	-2	-2	-2	-2	-1	-4	-				
M	1	-4	-4	1	-2	-2	-2	-2	-4	-1	-				
B	-4	-1	-1	-1	-1	-3	-3	-1	-1	-3	-				
V	-1	-4	-1	-1	-1	-3	-1	-3	-3	-1	-				
H	-1	-1	-4	-1	-3	-1	-3	-1	-3	-1	-				
D	-1	-1	-1	-4	-3	-1	-1	-3	-1	-3	-				
N	-2	-2	-2	-2	-1	-1	-1	-1	-1	-1	-				

Symbol	Meaning	Nucleic Acid
A	A	Adenine
C	C	Cytosine
G	G	Guanine
T	T	Thymine
U	U	Uracil
M	A or C	
R	A or G	purine
W	A or T	
S	C or G	
Y	C or T	pyrimidine
K	G or T	
V	A or C or G	not T
H	A or C or T	not G
D	A or G or T	not C
B	C or G or T	not A
X	G or A or T or C	
N	G or A or T or C	
.	G or A or T or C	