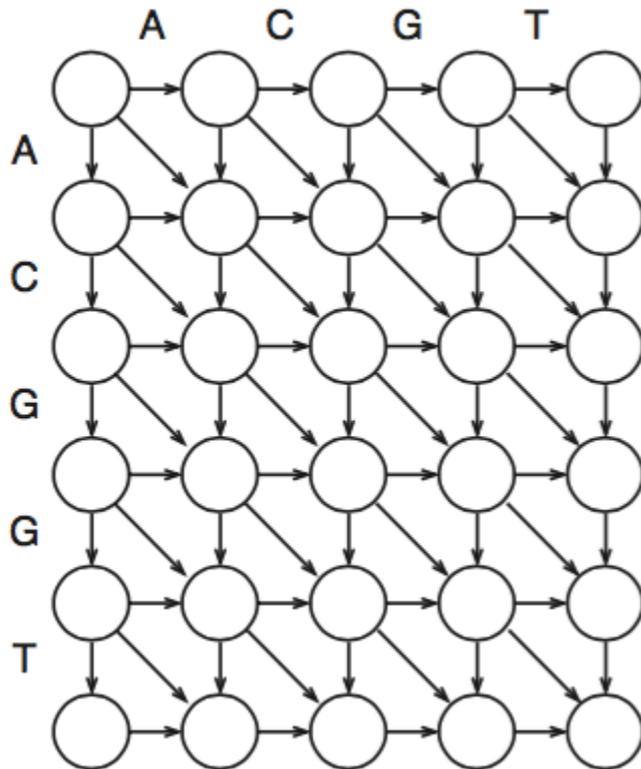


# Introducción a la Bioinformática

## Alineamiento de secuencias

## Búsqueda de secuencias en bases de datos

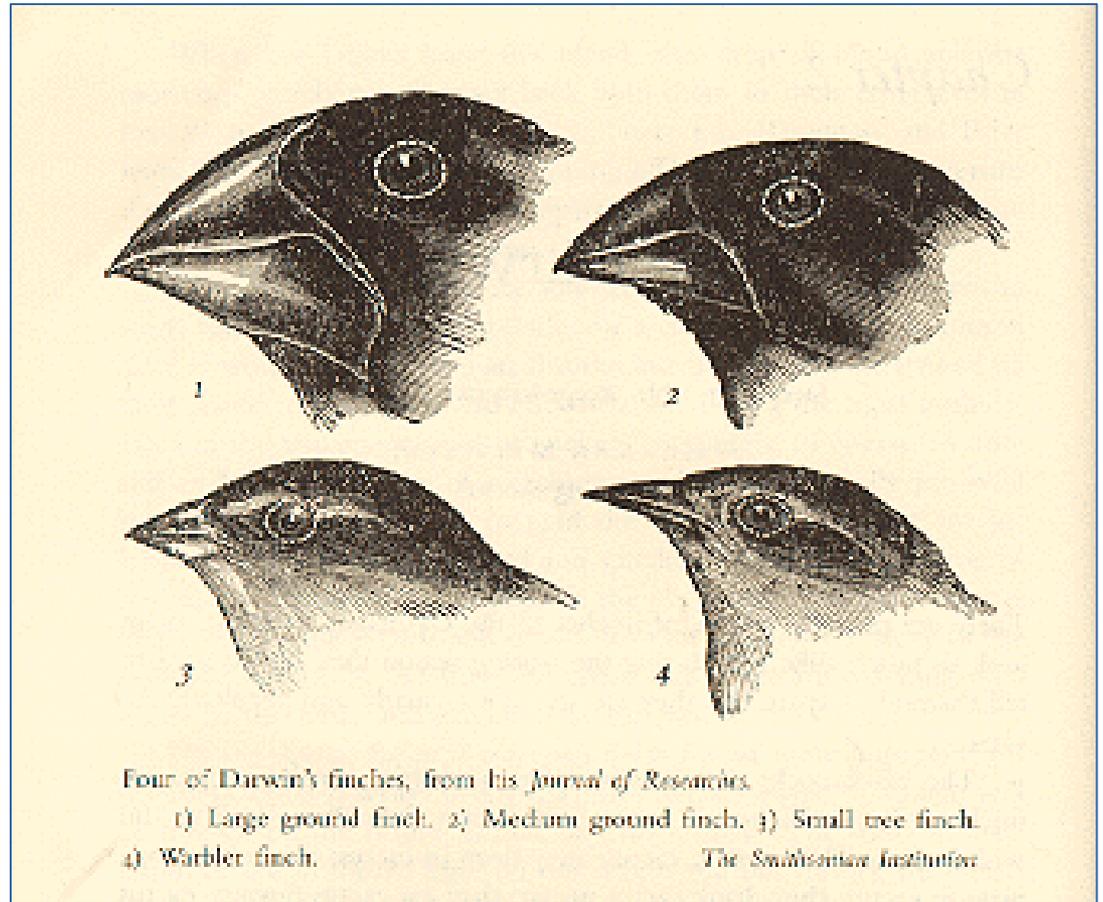


**Fernán Agüero**  
**Instituto de Investigaciones Biotecnológicas**  
**Universidad Nacional de San Martín**

# Análisis comparativo

El alineamiento de secuencias es similar a otros tipos de análisis comparativo.

En ambos es necesario cuantificar las similitudes y diferencias (scoring) entre un grupo relacionado de entidades.



Finches of the Galápagos Islands observed by Charles Darwin on the voyage of HMS *Beagle*

# Alinear secuencias

Para poder comparar secuencias, tenemos que sistematizar la manera en que lo hacemos

Por donde empezamos?

Comparamos las dos secuencias letra a letra, empezando por la primera?

Tiene sentido?

**GCTACTAGTTCGCTTAGC**

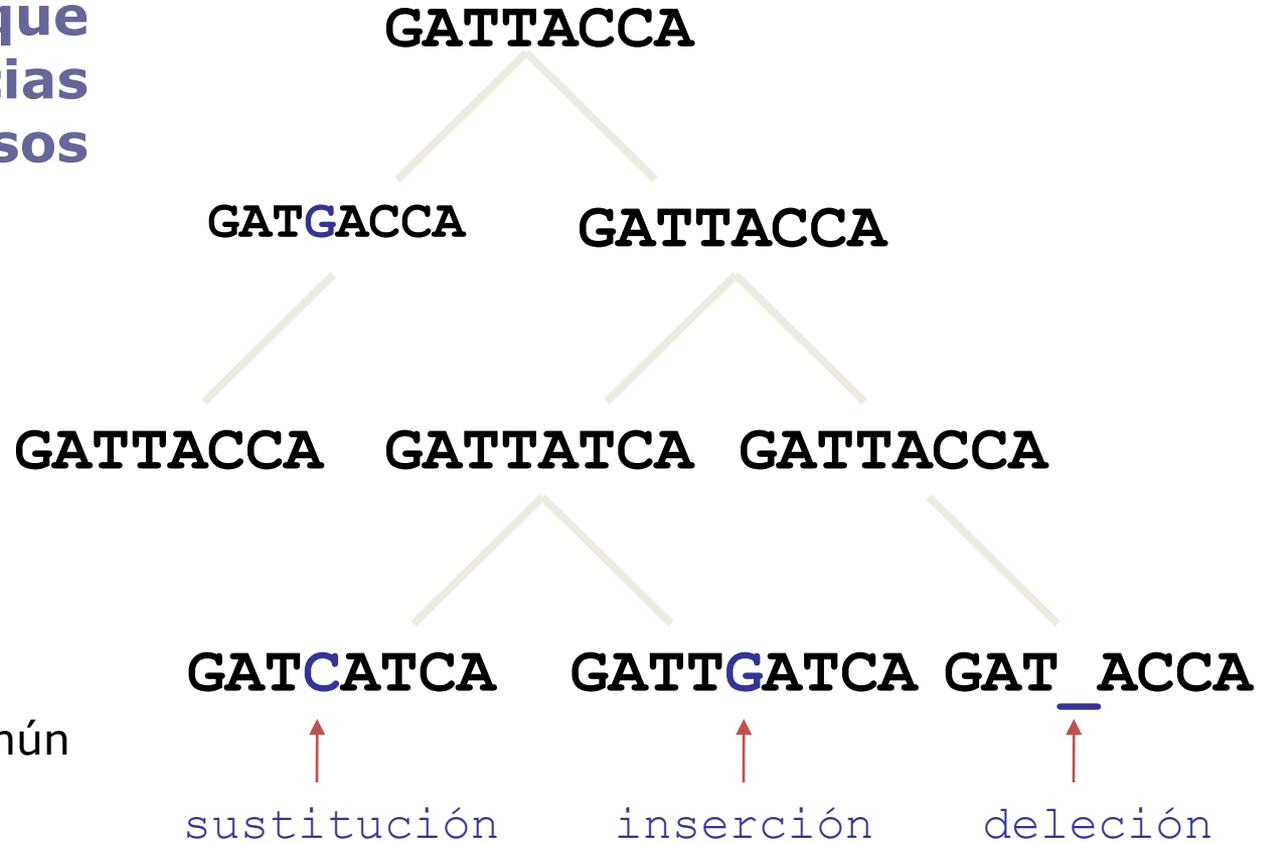
**GCTACTAGCTCTAGCGCGTATAGC**

# Homología vs similitud

- **Homología entre dos entes biológicos implica una herencia compartida**
- **Homología es un término cualitativo**
- **Se es homólogo o no se es**
- **Similitud implica una apreciación cuantitativa o una cuantificación directa de algún carácter**
- **Podemos usar una medida de similitud para **inferir** homología**

# Análisis comparativo

Los algoritmos que alinean secuencias modelan procesos evolutivos

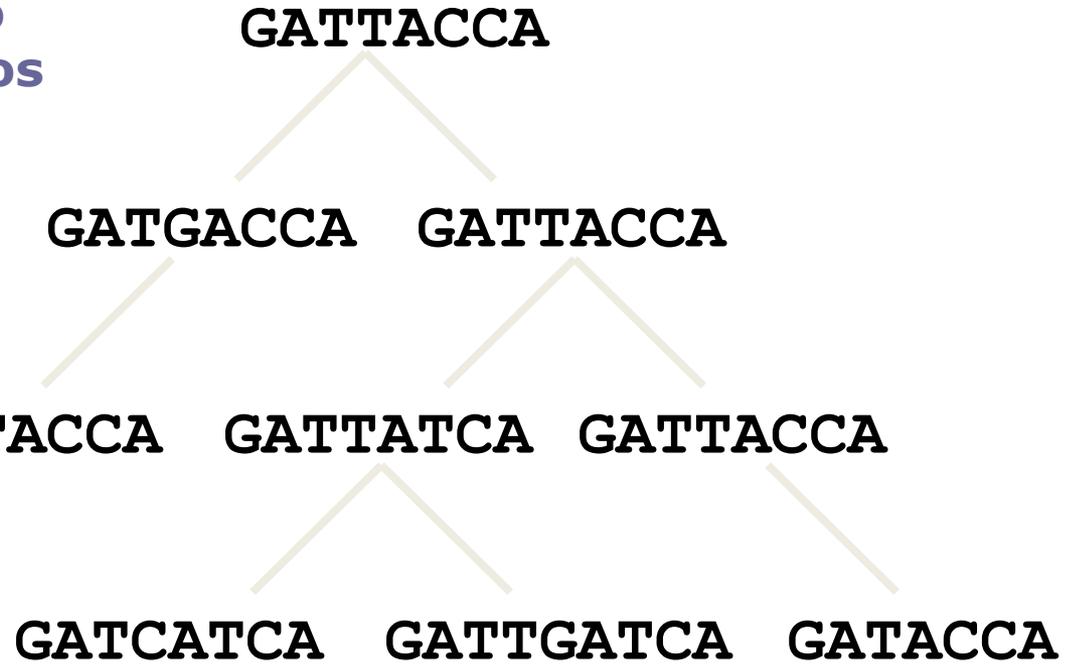


Deriva de un ancestro común a través de cambios incrementales debido a errores en la replicación del DNA, mutaciones, daño o crossing-over desigual.

# Análisis comparativo

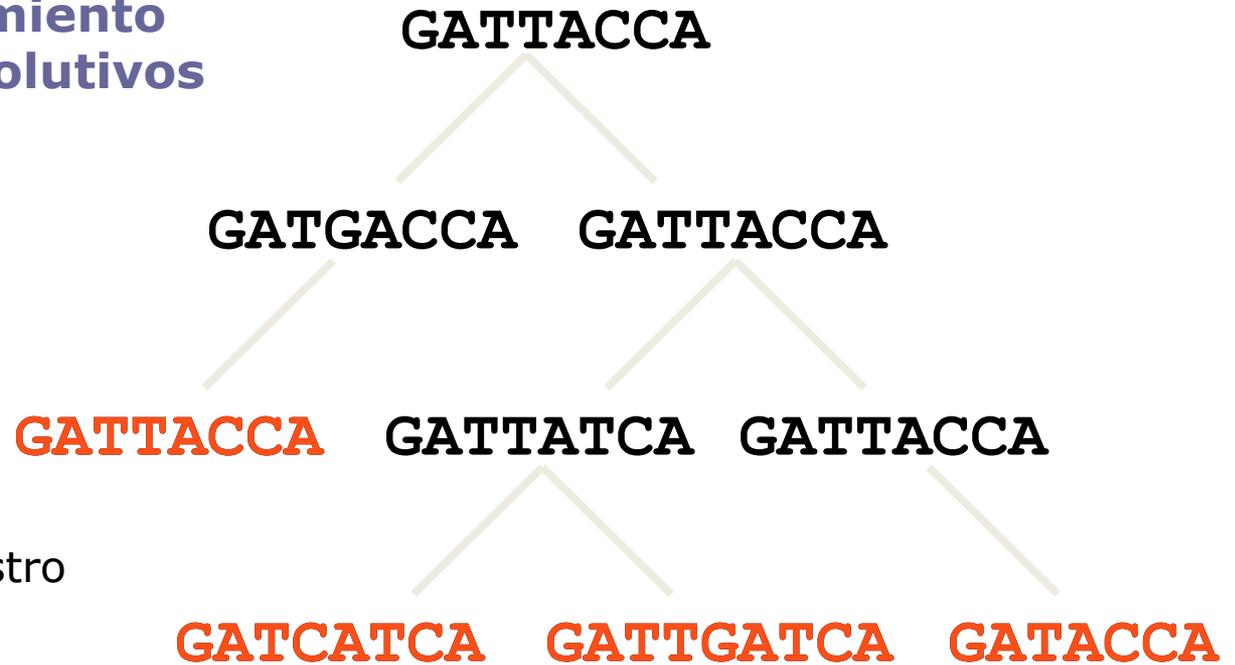
Algoritmos de alineamiento modelan procesos evolutivos

Deriva a partir de un ancestro común a través de cambio incremental.



# Análisis comparativo

Algoritmos de alineamiento modelan procesos evolutivos

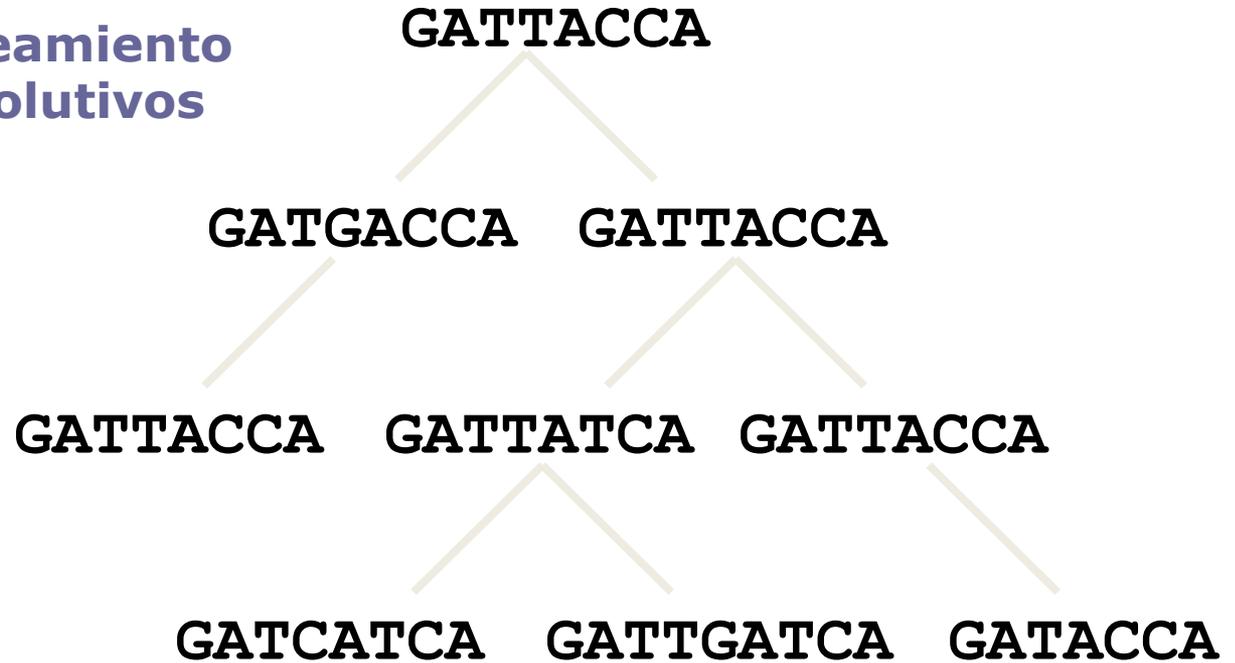


Deriva a partir de un ancestro común a través de cambio incremental.

**Sólo las secuencias actuales son conocidas, las secuencias ancestrales se postulan.**

# Análisis comparativo

Algoritmos de alineamiento modelan procesos evolutivos



Deriva a partir de un ancestro común a través de cambio incremental. Mutaciones que no matan al individuo pueden pasar a la población.

La palabra **homología** implica una herencia común (un ancestro común), el cual puede ser inferido a partir de observaciones de **similitud** de secuencia.

- **Qué es un alineamiento?**
  - **El procedimiento de comparación de dos (o más) secuencias que busca una serie de caracteres individuales o patrones de caracteres que se encuentren en el mismo orden en ambas secuencias**
- **Cómo alineamos dos secuencias?**
  - **Usando un método (algoritmo)**
    - a mano (como en los viejos tiempos)
    - usando una computadora

# Definición de alineamiento: tipos

- Alineamiento:** Cada base se usa a lo sumo una vez
- Alineamiento global:** Todas las bases se alinean con otra base o con un gap ("-")
- Alineamientos locales:** No hay necesidad de alinear todas las bases

Align **GATESLIKESCHEESE** and **GRATEDCHEESE**

<b>G-ATESLIKESCHEESE</b>	<b>or</b>	<b>G-ATES</b>	<b>&amp;</b>	<b>CHEESE</b>
<b>GRATED-----CHEESE</b>		<b>GRATED</b>	<b>&amp;</b>	<b>CHEESE</b>

# Alineamientos buenos y malos?

Cuál es el 'mejor' alineamiento?

GCTACTAG-T-T--CGC-T-TAGC  
GCTACTAGCTCTAGCGCGTATAGC

0 mismatches, 5 gaps

GCTACTAGTT-----CGCTTAGC  
GCTACTAGCTCTAGCGCGTATAGC

3 mismatches, 1 gap

# Cómo decidir cuál es el mejor?

- Pero: necesitamos una medida **objetiva**
- **sistemas de puntaje (scoring)**
  - reglas para asignar puntos
  - el más simple: match, mismatch, gap

# Cómo decidir cuál es el mejor?

- **Respuesta: el más significativo desde el punto de vista biológico**
- **Pero: necesitamos una medida **objetiva****
- **sistemas de puntaje (scoring)**
  - **reglas para asignar puntos**
  - **el más simple: match, mismatch, gap**

# Cómo decidir cuál es el mejor?

- **Respuesta: el más significativo desde el punto de vista biológico**
- **Pero: necesitamos una medida **objetiva****
- **sistemas de puntaje (scoring)**
  - **reglas para asignar puntos**
  - **el más simple: match, mismatch, gap**

# Cómo decidir cuál es el mejor?

- **Respuesta: el más significativo desde el punto de vista biológico**
- **Pero: necesitamos una medida **objetiva****
- **sistemas de puntaje (scoring)**
  - **reglas para asignar puntos**
  - **el más simple: match, mismatch, gap**

# Un primer sistema de puntajes

## Ejemplo de sistema de score

**match = +1**

**mismatch = 0**

**gap = -1**

G-ATESLIKESCHEESE  
GRATED-----CHEESE

# Un primer sistema de puntajes

## Ejemplo de sistema de score

match = +1

mismatch = 0

gap = -1

G-ATESLIKESCHEESE  
GRATED-----CHEESE

### Score

$$(10 * 1) + (1 * 0) + (6 * (-1)) = +4$$

# Cambiamos nuestro sistema de puntajes

## Usando otro de sistema de puntajes?

**match = +2**

**mismatch = 0**

**gap = -1**

**G-ATESLIKESCHEESE**  
**GRATED-----CHEESE**

## Usando otro sistema de score

$$\text{Score} \\ (10 * 2) + (1 * 0) + (5 * (-1)) = +14$$

# No se pueden comparar scores

- **no tiene sentido comparar scores de distintos alineamientos**
- **a menos que se especifique el sistema de scoring utilizado**

# No se pueden comparar scores

- **Primera conclusión importante:**
  - **no tiene sentido comparar scores de distintos alineamientos**
  - **a menos que se especifique el sistema de scoring utilizado**

# No se pueden comparar scores

- **Primera conclusión importante:**
  - **no tiene sentido comparar scores de distintos alineamientos**
  - **a menos que se especifique el sistema de scoring utilizado**

# No se pueden comparar scores

- **Primera conclusión importante:**
  - **no tiene sentido comparar scores de distintos alineamientos**
  - **a menos que se especifique el sistema de scoring utilizado**

# Gap penalties

gap opening penalty = -5

gap extension penalty = -1

1- Abrir un gap es costoso

GCTACTAG-T-T--CGC-T-TAGC  
GCTACTAGCTCTAGCGCGTATAGC

$$\text{Penalty} = 5 * (-5) + 6 * (-1) = -31$$

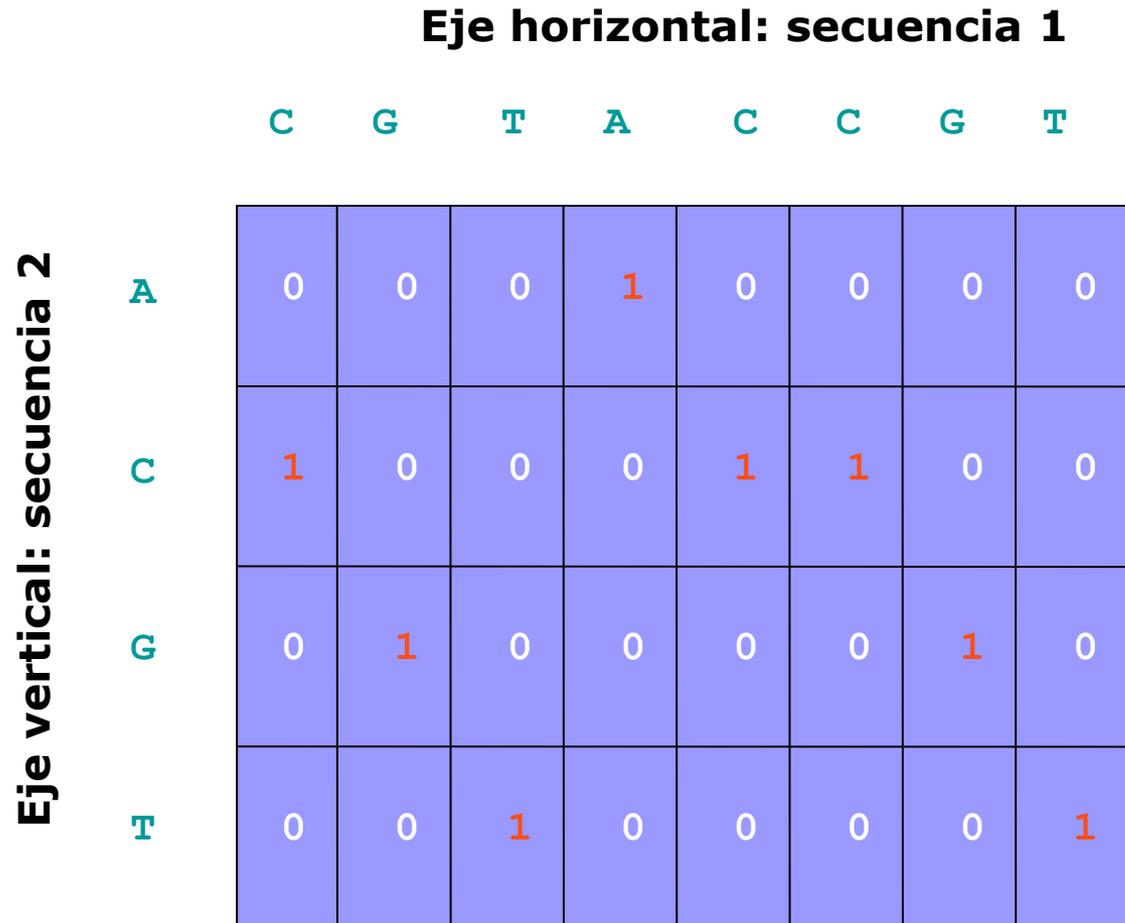
2 - Extender un gap es menos costoso

GCTACTAGTT-----CGCTTAGC  
GCTACTAGCTCTAGCGCGTATAGC

$$\text{Penalty} = 1 * (-5) + 6 * (-1) = -11$$

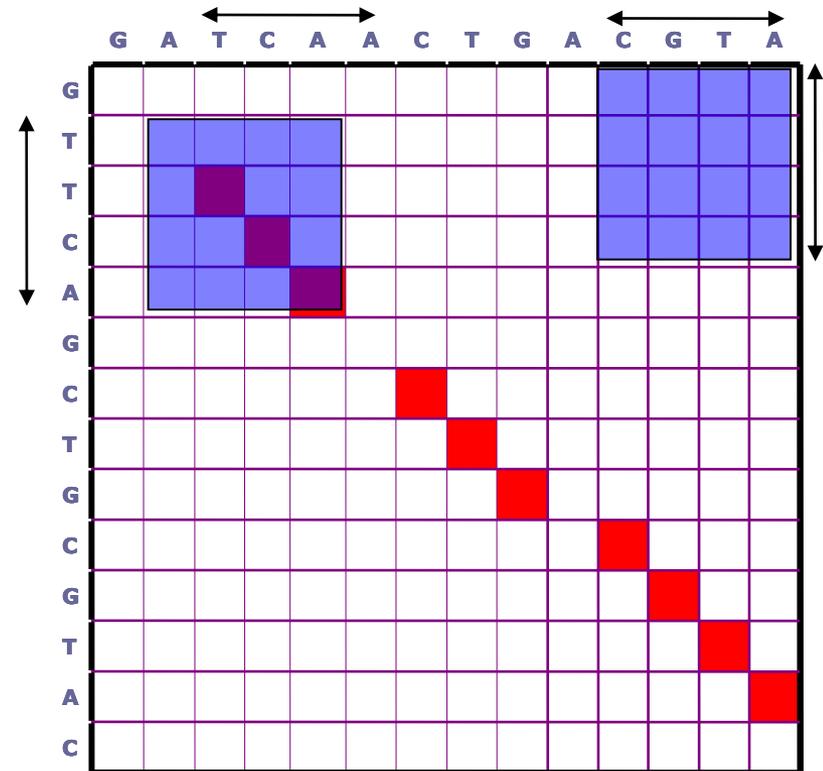
# Dot plots: introducción

Dot-plot: Fitch, Biochem. Genet. (1969) 3, 99-108.

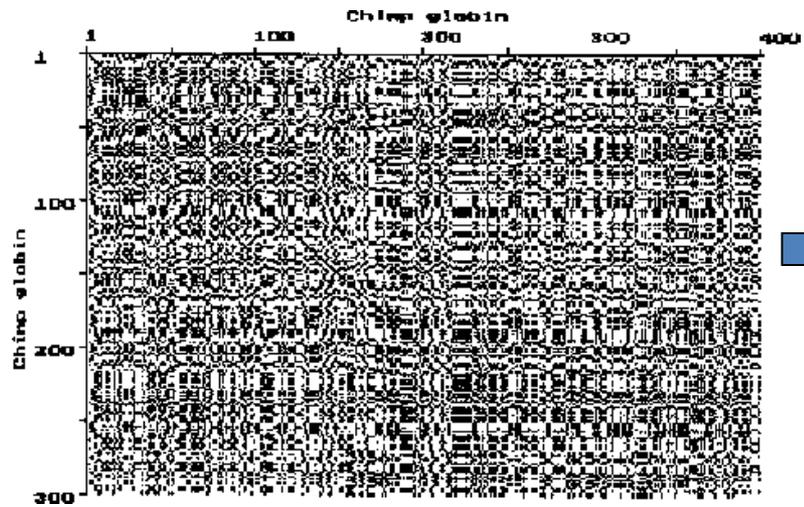


# Dot Matrix Plot

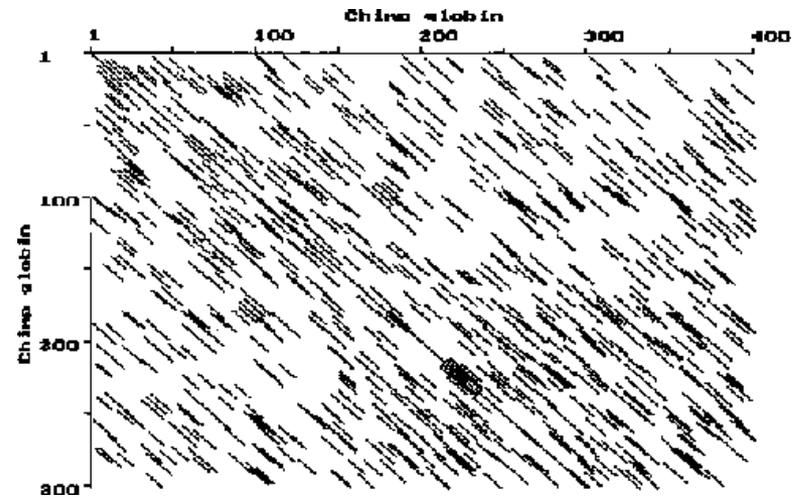
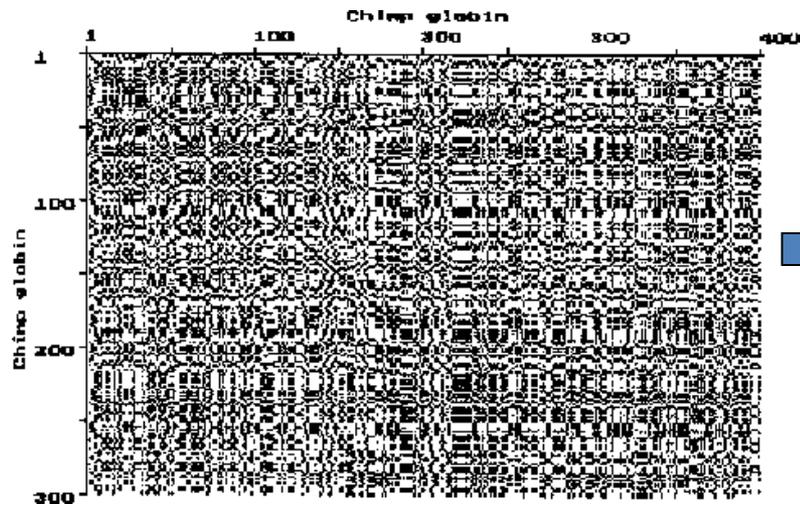
- Dos secuencias, una vertical y otra horizontal a los ejes del gráfico.
- Se colocan “puntos” en donde hay un match.
- Las líneas diagonales son regiones de identidad.
- Se aplican filtros para mejorar la comprensión del gráfico.



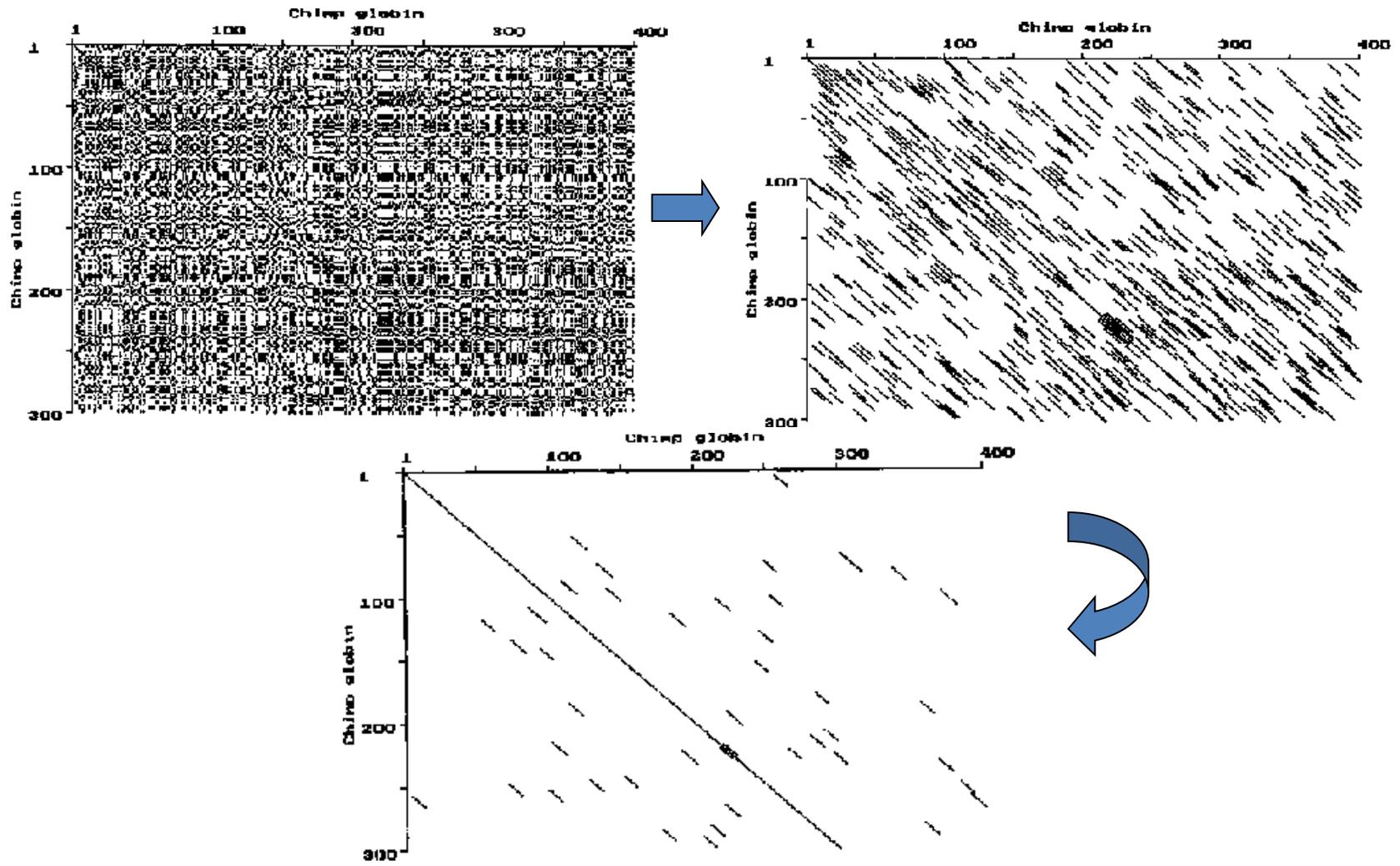
# Dot Matrix Plot



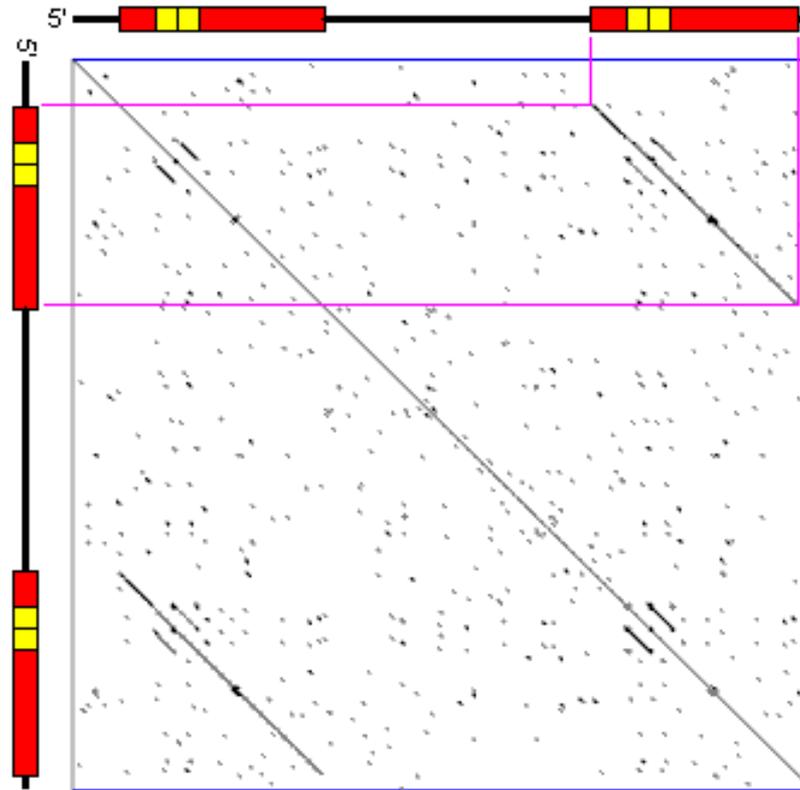
# Dot Matrix Plot



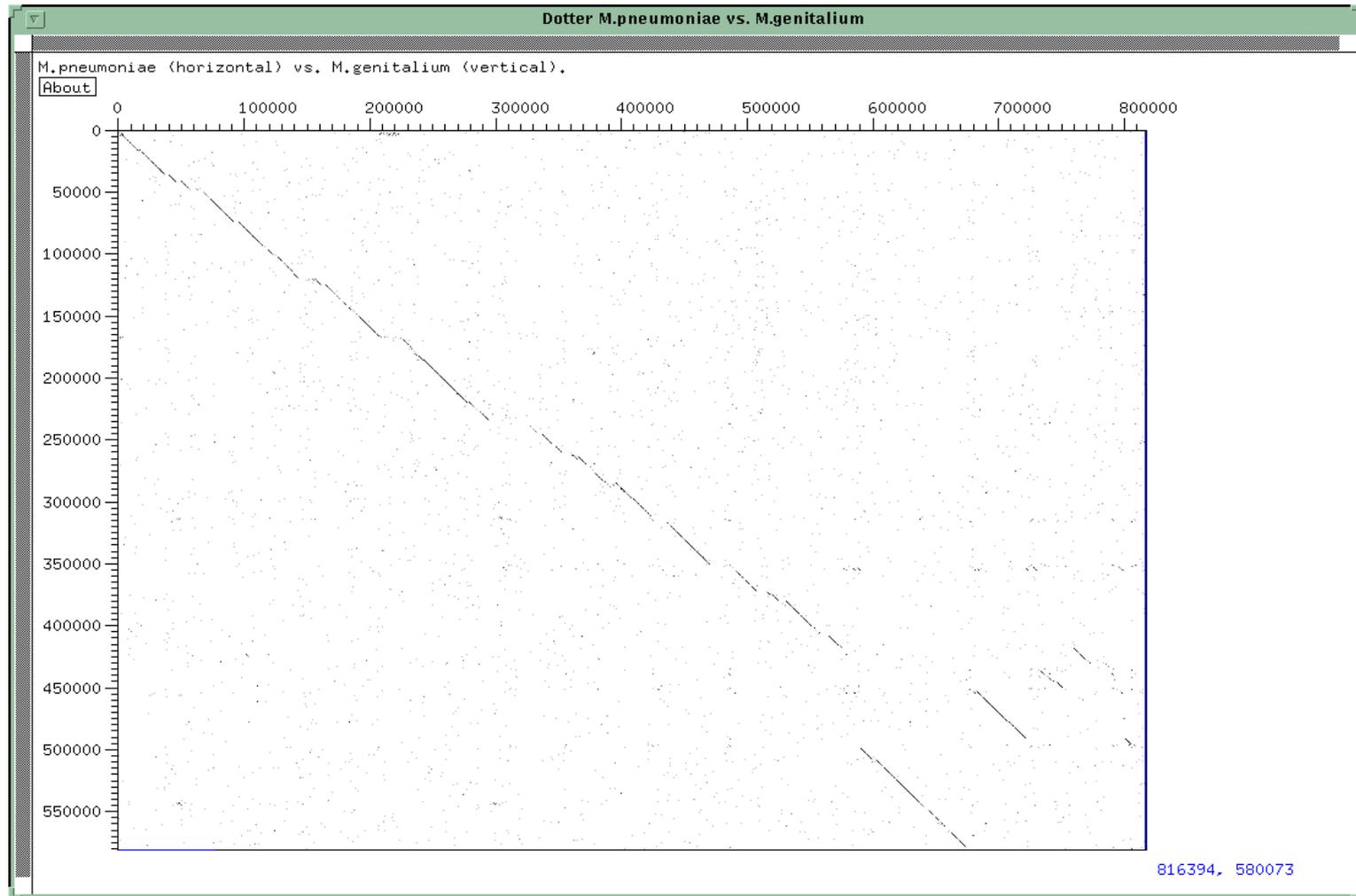
# Dot Matrix Plot



# Dot Matrix Plot



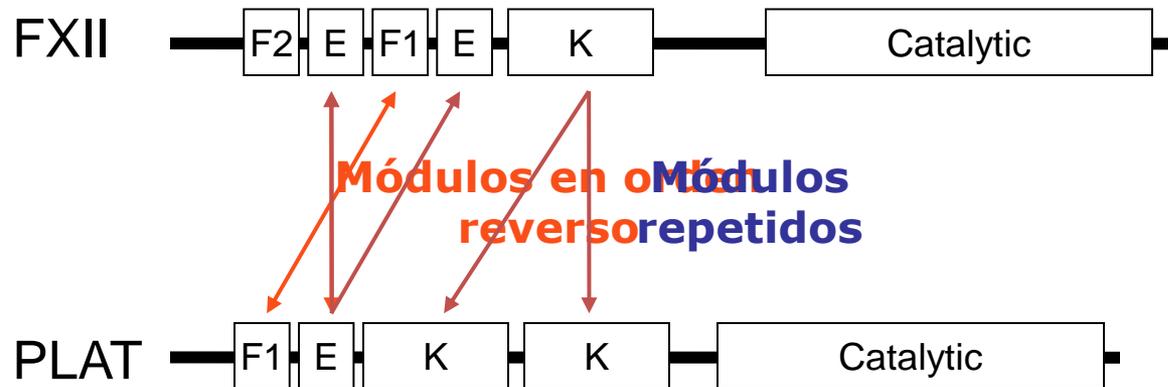
# Dot Matrix Plot



# Similitud local

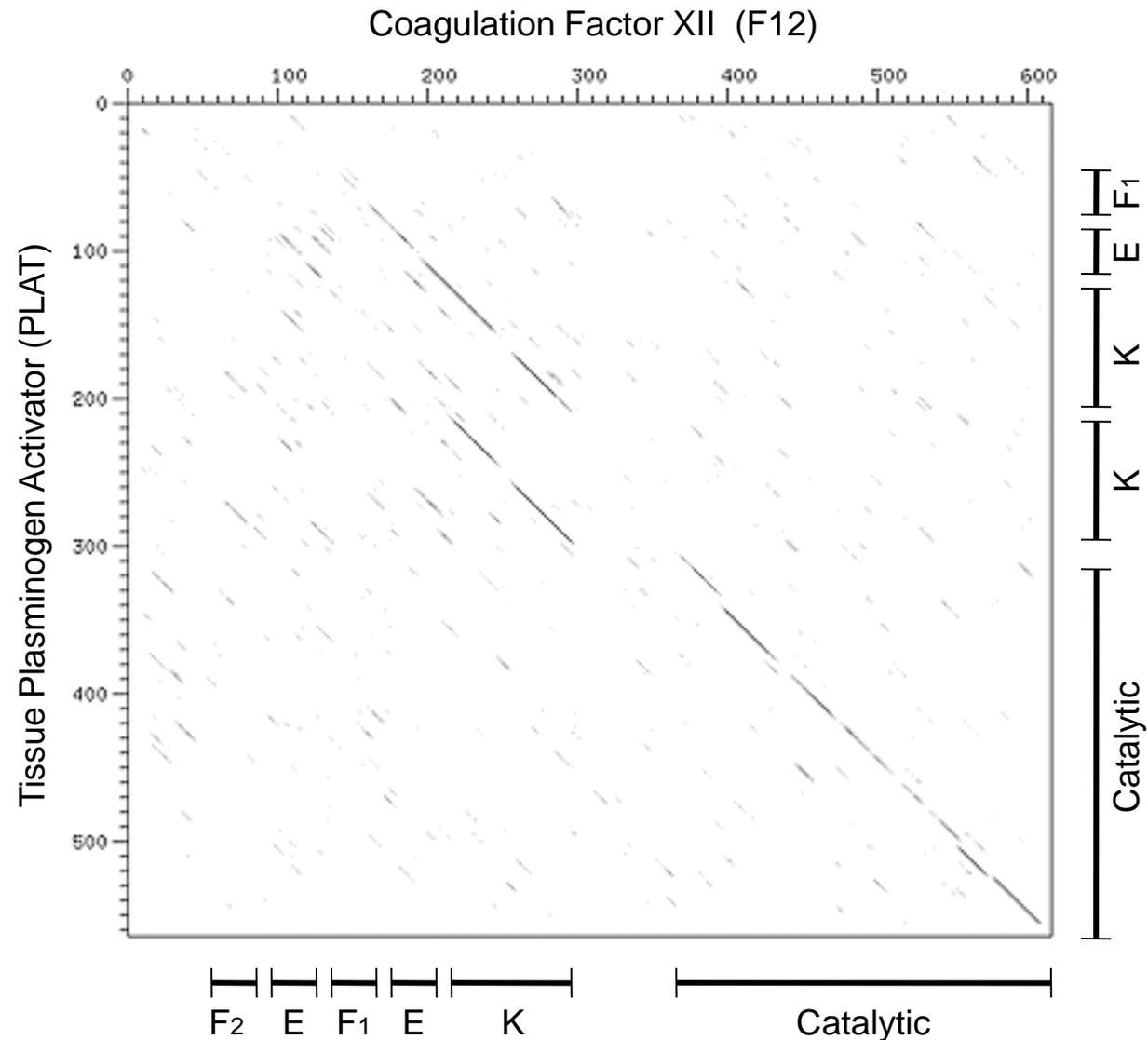
Dominios mezclados confunden a los algoritmos de alineamiento.

Módulos en el factor XII de coagulación y en el activador de plasminógenos – tissue plasminogen activator (PLAT)



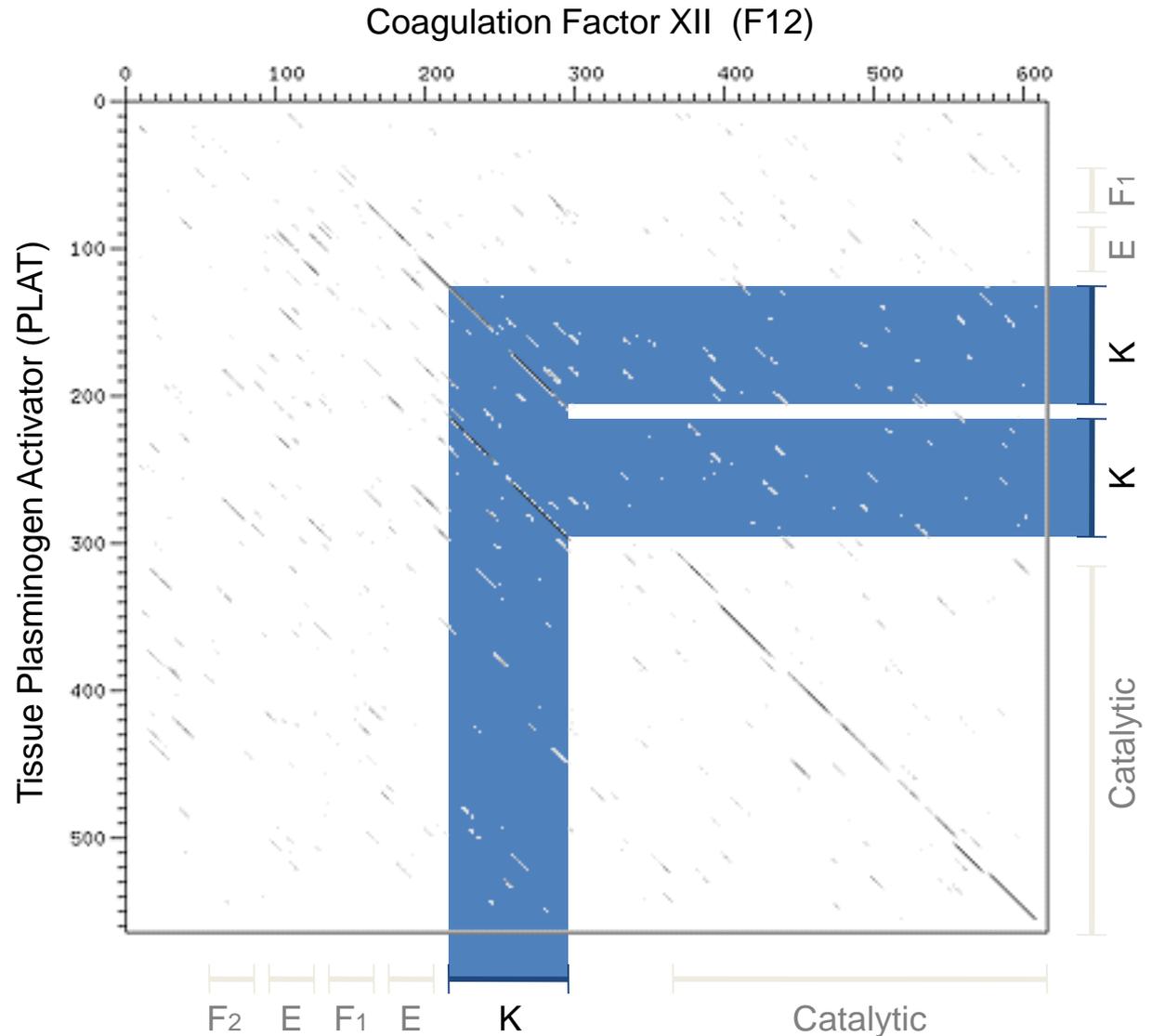
F1,F2	Fibronectin repeats
E	EGF similarity domain
K	Kringle domain
Catalytic	Serine protease activity

# Dot plots: ejemplo



# Dot plots: ejemplo (cont.)

Dominios repetidos muestran un patrón característico.



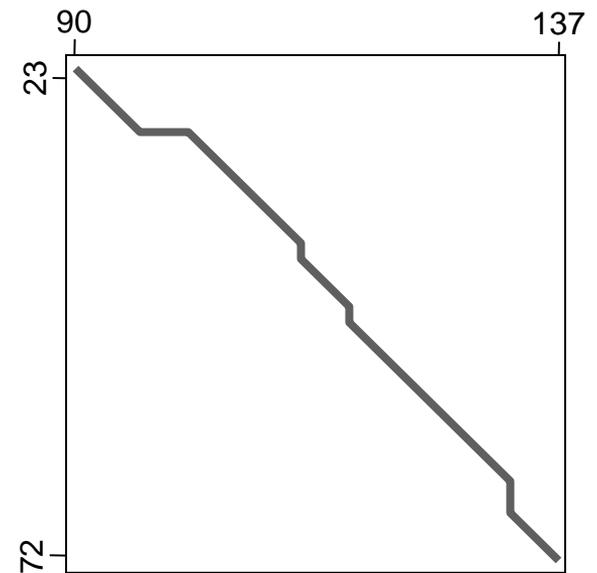
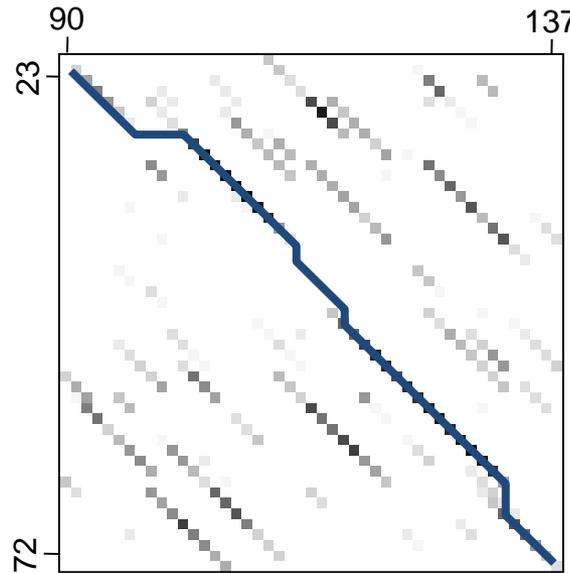
# Dot plots: path graphs

Dot plots sugieren caminos (paths) a través del espacio de alineamientos posibles.

Path graphs son representaciones más explícitas de un alineamiento.

Cada path es un alineamiento único.

**Dominios EGF conservados en la urokinase plasminogen activator (PLAU) y el tissue plasminogen activator (PLAT)**



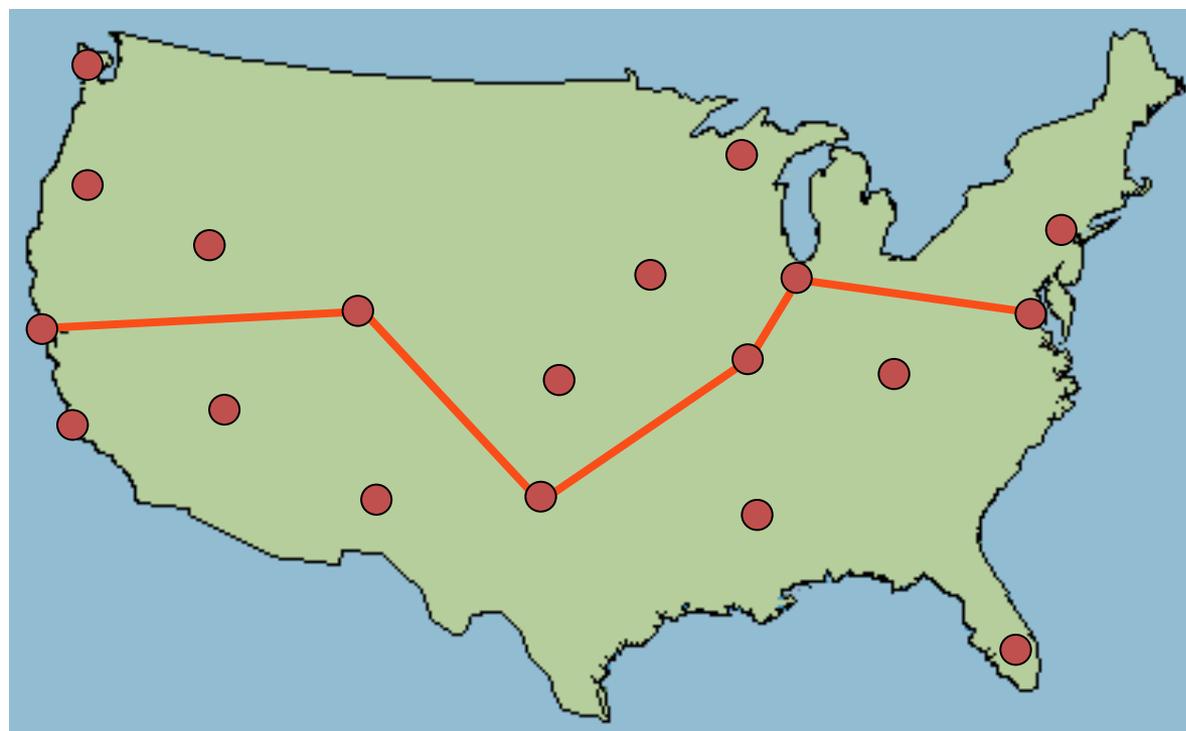
PLAU	90	EPKKVKDHC SKHSPCQKGGTCVNMP--SGPH-CLCPQH LTGNHCQKEK---CFE	137
PLAT	23	ELHQVPSNCD----CLNGGTCVSNKYFSNIHWCNCPKKFGGQHCEIDKSKTCYE	72

# Path graphs: encontrar el mejor camino

Los problemas que involucran encontrar la mejor ruta o camino (Best-path problems) son comunes en computación científica.

El algoritmo para encontrar el mejor camino entre dos extremos y pasando por varios puntos se llama 'dynamic programming'

## Rutear una llamada telefónica desde NY a San Francisco



# Dynamic programming: introducción

## Un ejemplo:

Construir un  
alineamiento óptimo  
entre estas dos  
secuencias

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

Match: +1

Mismatch: -1

Gap: -1

# Dynamic programming: introducción

## Un ejemplo:

Construir un  
alineamiento óptimo  
entre estas dos  
secuencias

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

Utilizando las  
siguientes reglas de  
scoring:

```
Match:      +1
Mismatch:   -1
Gap:        -1
```

# Dynamic programming: ejemplo

Ordenar las dos  
secuencias en una  
matriz bidimensional

Los vértices de cada  
celda se encuentran  
entre letras (bases).

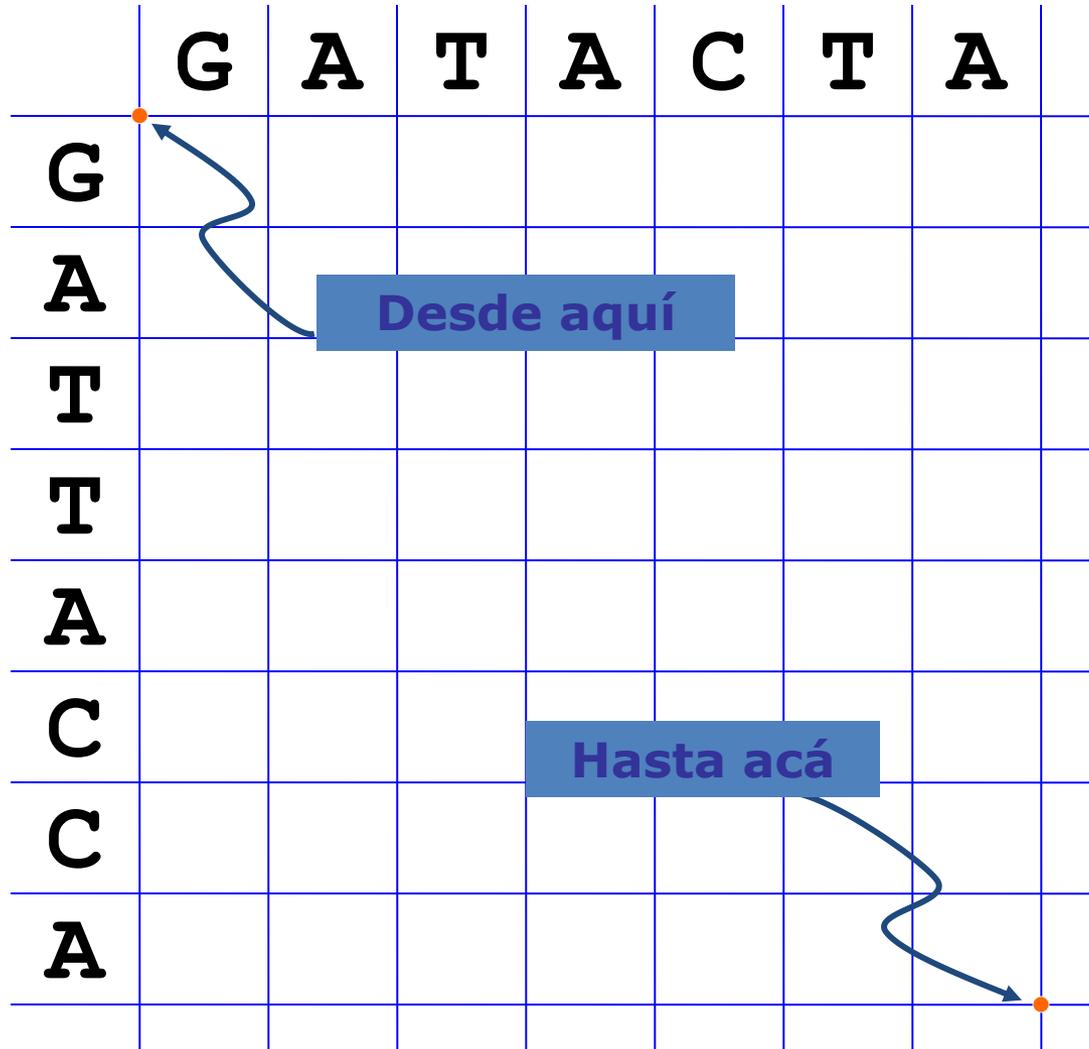
Needleman & Wunsch  
(1970)

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

Slides Dynamic Programming: Hugues Sicotte (NCBI)

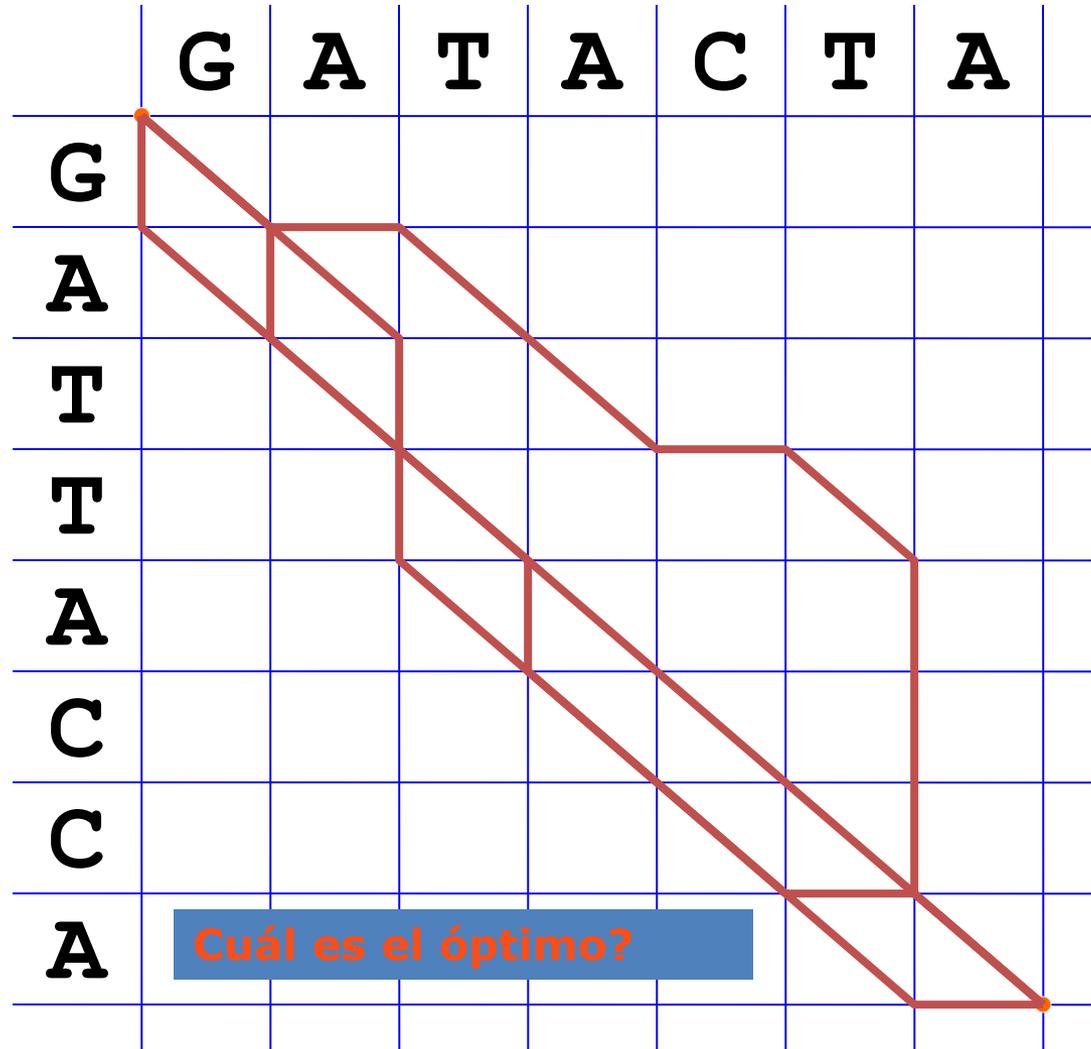
# Dynamic programming: ejemplo (cont.)

El objetivo es encontrar la ruta (path) óptimo



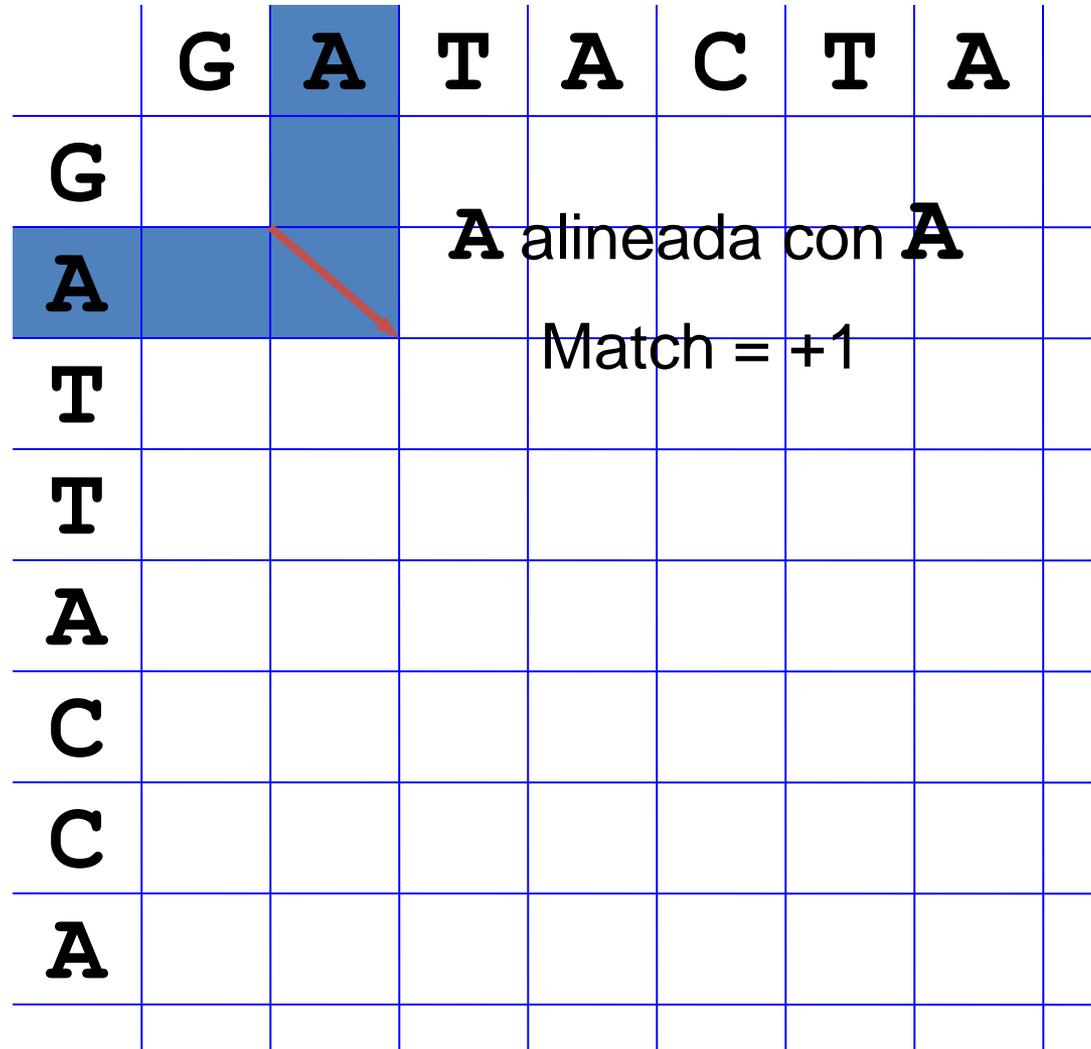
# Dynamic programming: paths posibles

Cada path corresponde a un alineamiento único



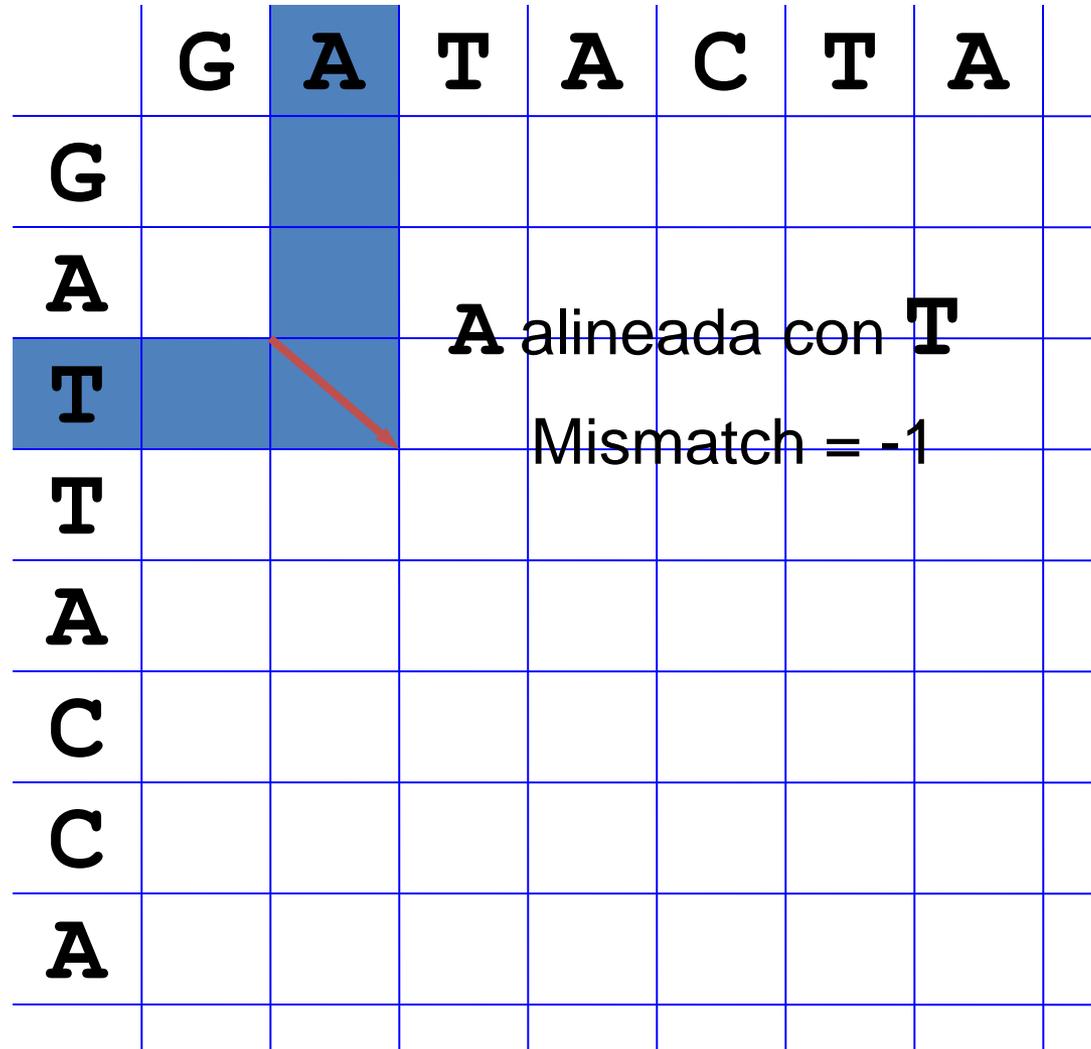
# Dynamic programming: scores: match

El score para una ruta (path) es la suma incremental de los scores de sus pasos (diagonales o lados).



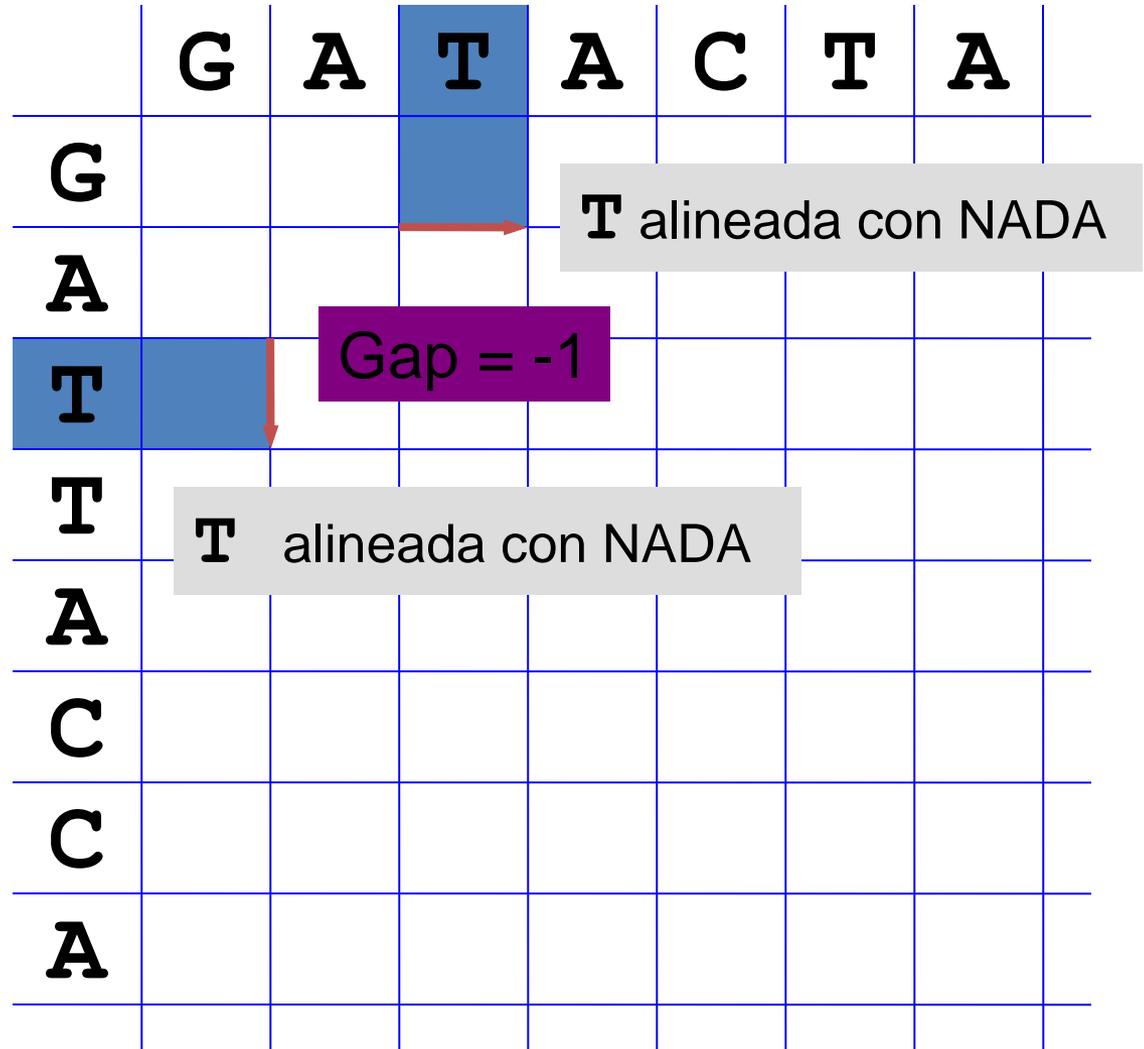
# Dynamic programming: scores: mismatch

El score para una ruta (path) es la suma incremental de los scores de sus pasos (diagonales o lados).



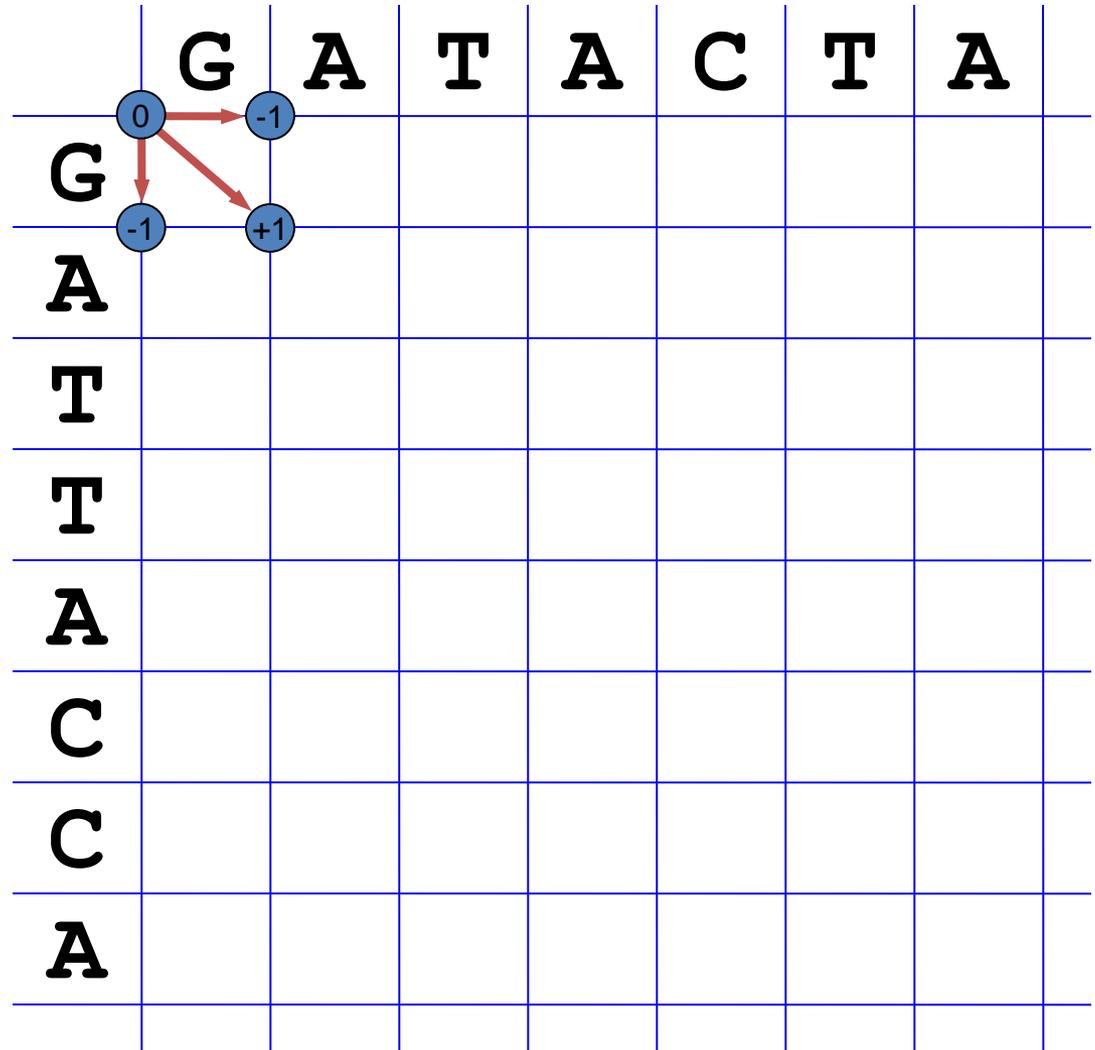
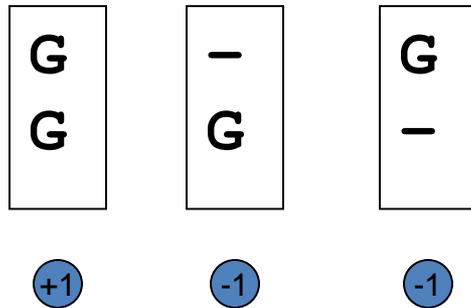
# Dynamic programming: scores: gaps

El score para una ruta (path) es la suma incremental de los scores de sus pasos (diagonales o lados).



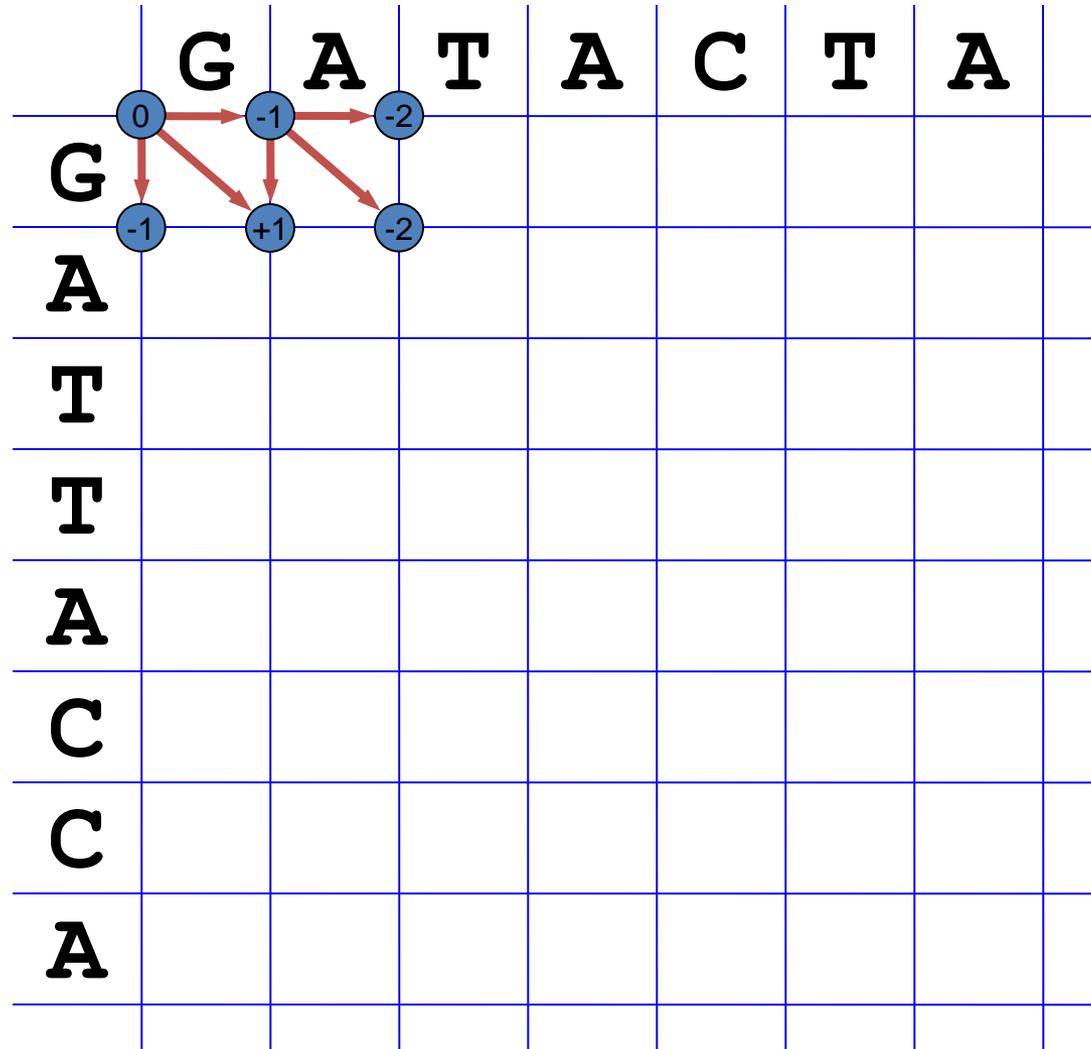
# Dynamic programming: paso a paso (1)

## Extender el path paso por paso



# Dynamic programming: paso a paso (2)

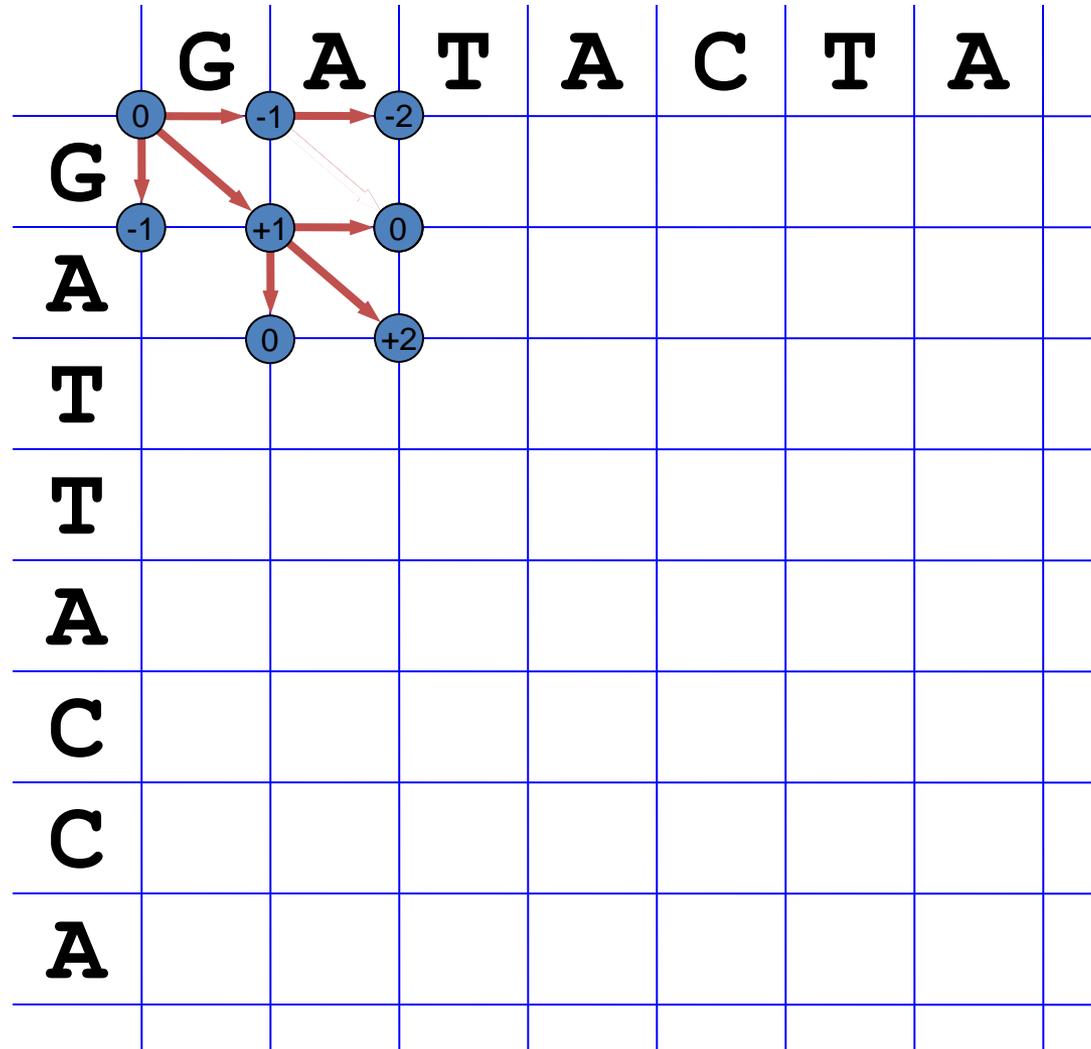
Incrementar el path paso a paso



Recordar el mejor subpath que lleva a cada punto en la matriz.

# Dynamic programming: paso a paso (3)

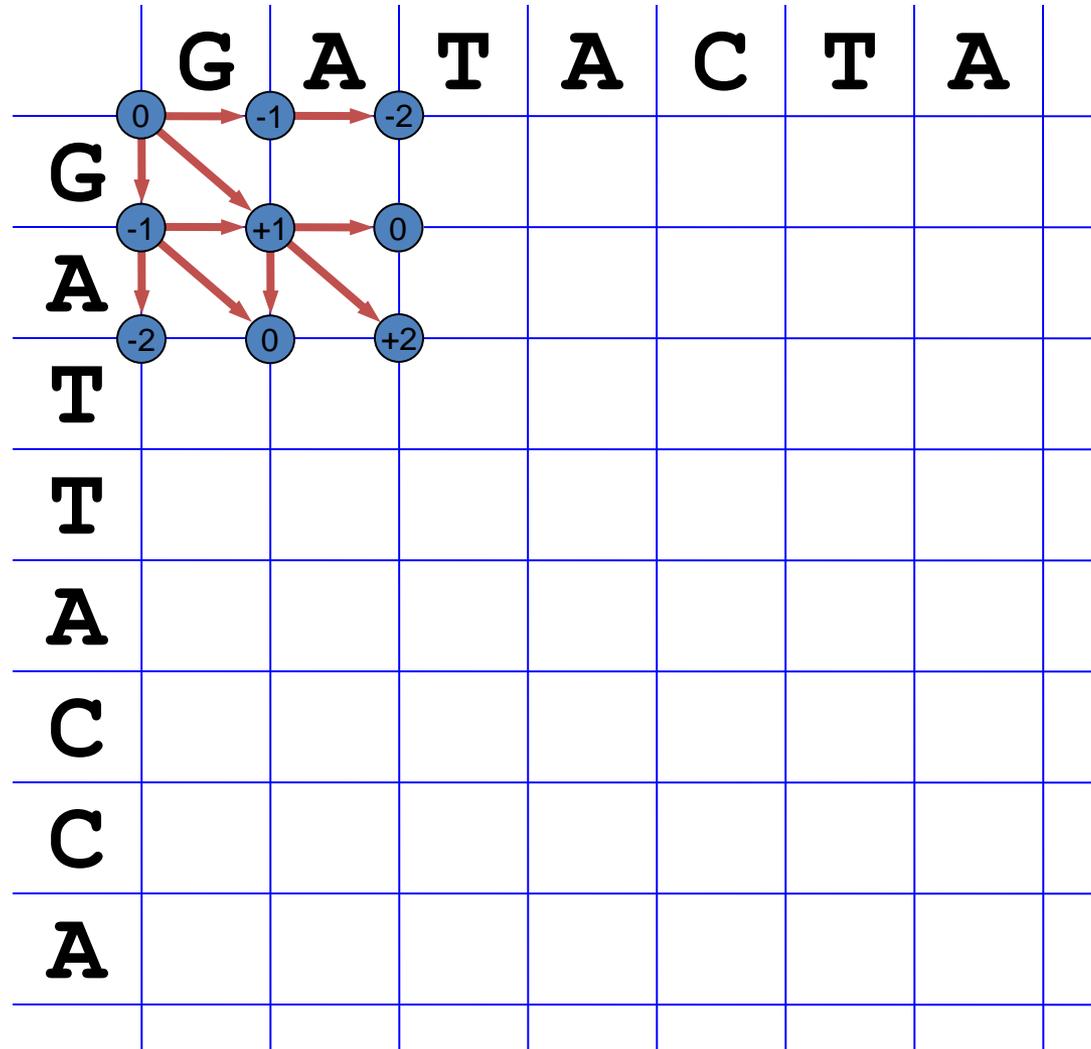
## Incrementar el path paso a paso



Recordar el mejor subpath que lleva a cada punto en la matriz.

# Dynamic programming: paso a paso (4)

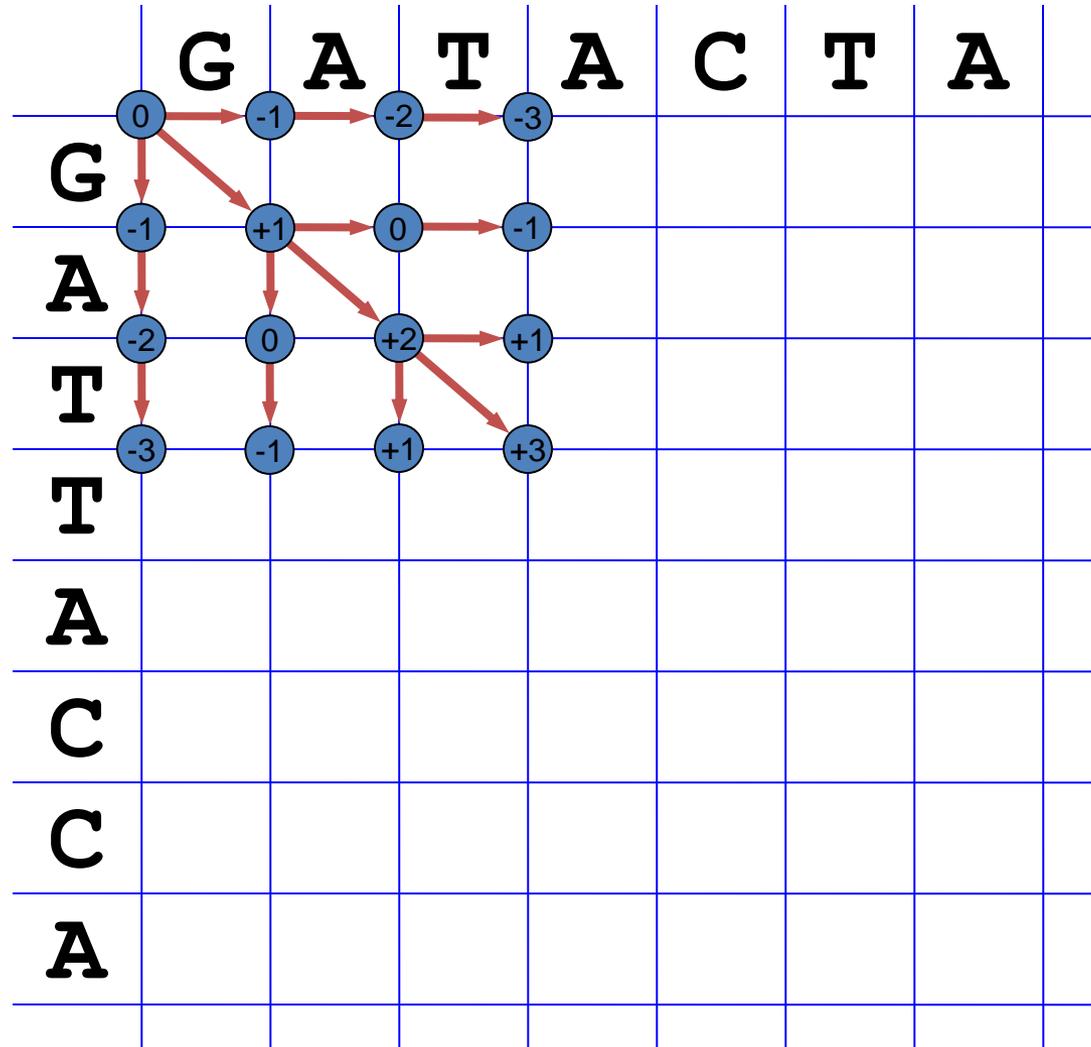
## Incrementar el path paso a paso



Recordar el mejor subpath que lleva a cada punto en la matriz.

# Dynamic programming: paso a paso (5)

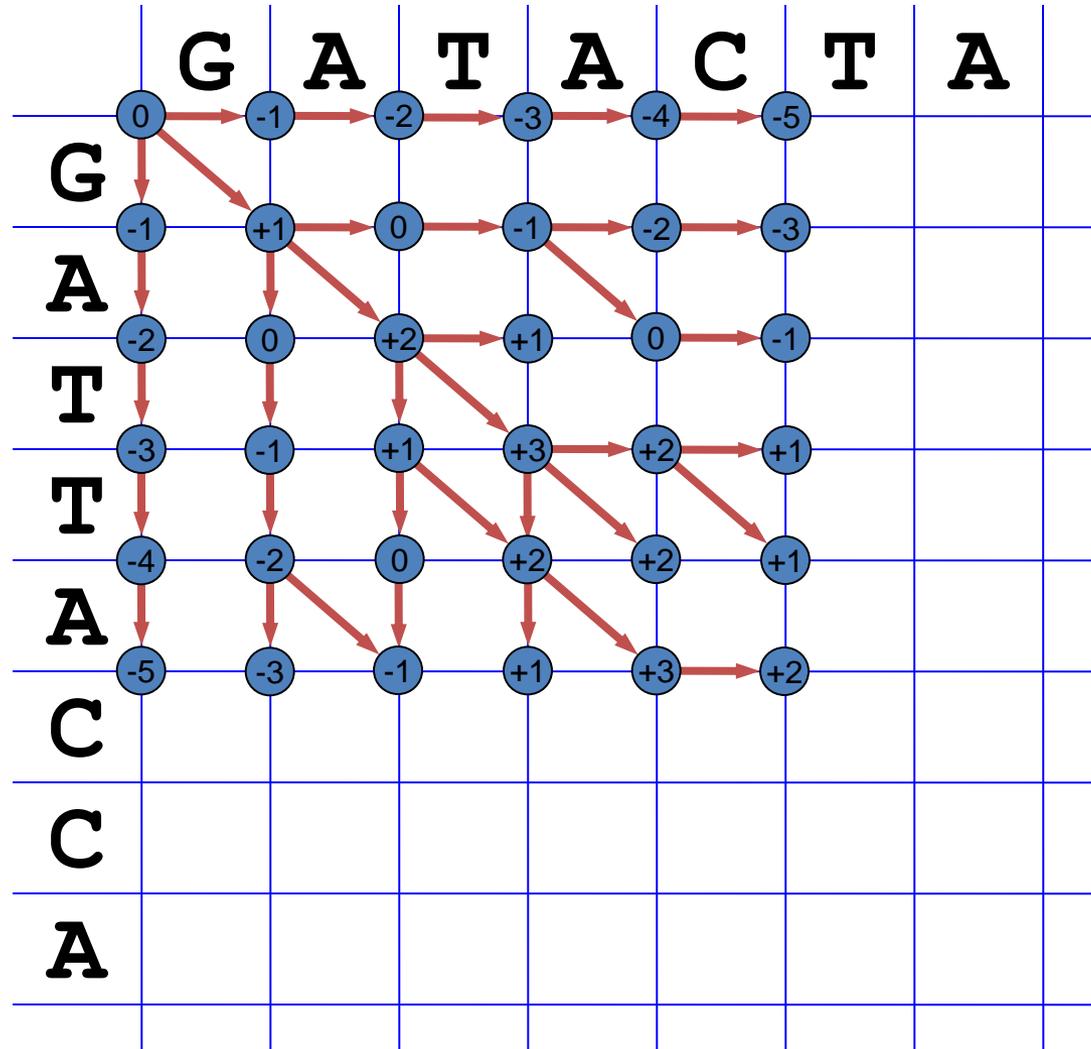
## Incrementar el path paso a paso



Recordar el mejor subpath que lleva a cada punto en la matriz.

# Dynamic programming: paso a paso (6)

## Incrementar el path paso a paso

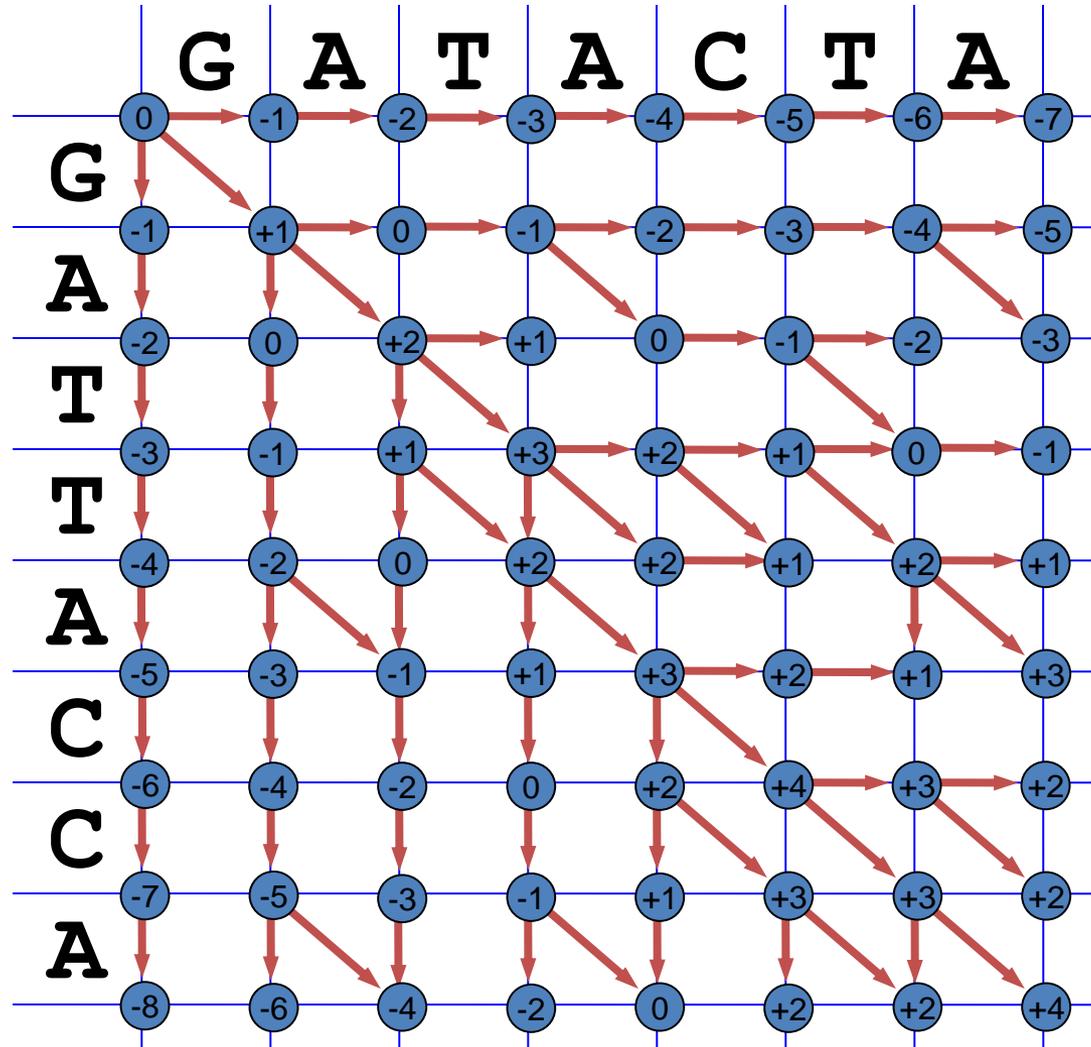


Recordar el mejor subpath que lleva a cada punto en la matriz.

# Dynamic programming: paso a paso (7)

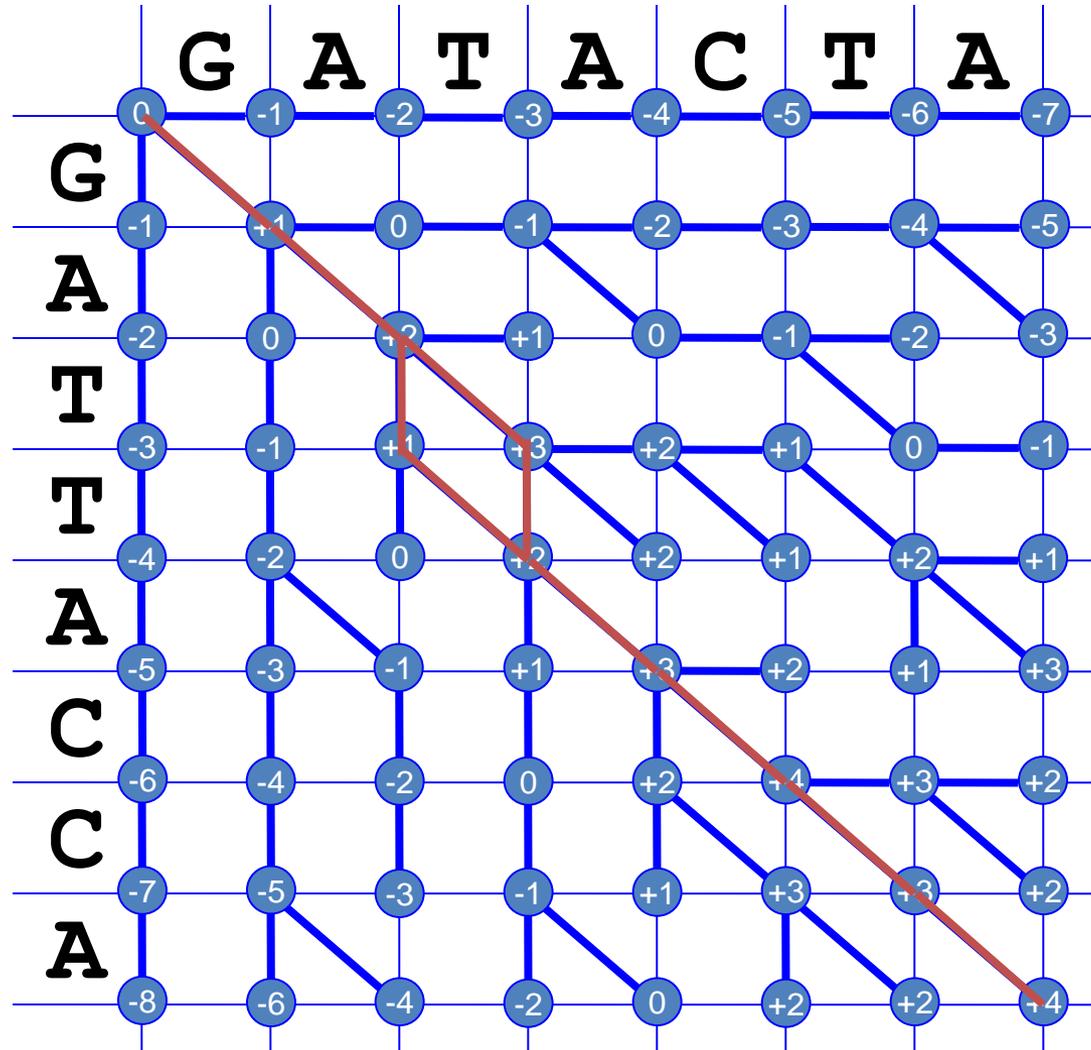
## Incrementar el path paso a paso

Recordar el mejor subpath que lleva a cada punto en la matriz.



# Dynamic programming: best path

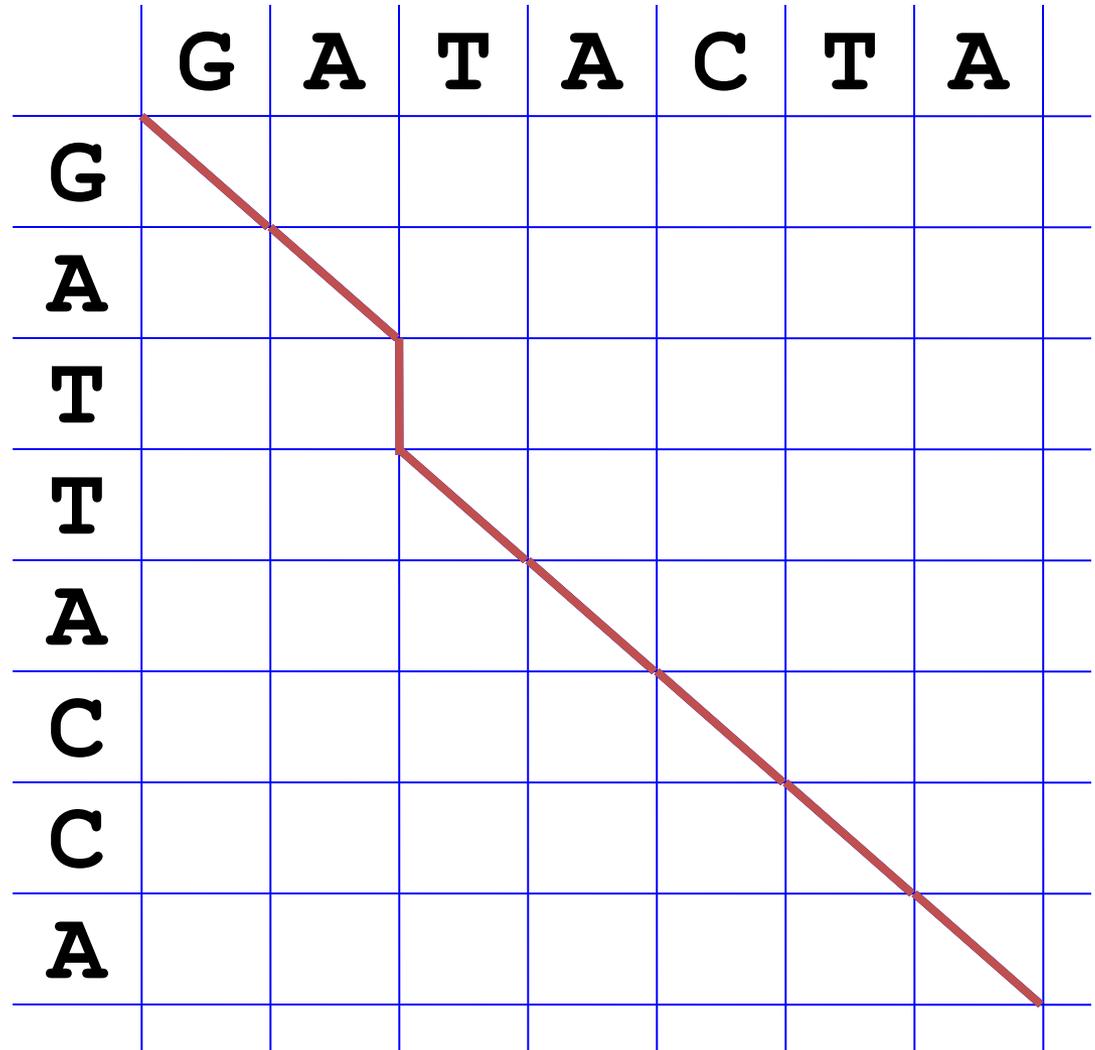
Recorrer el camino de atrás hacia adelante para obtener el mejor path y alineamiento.



# Dynamic programming: alineamiento obtenido

Imprimir el alineamiento

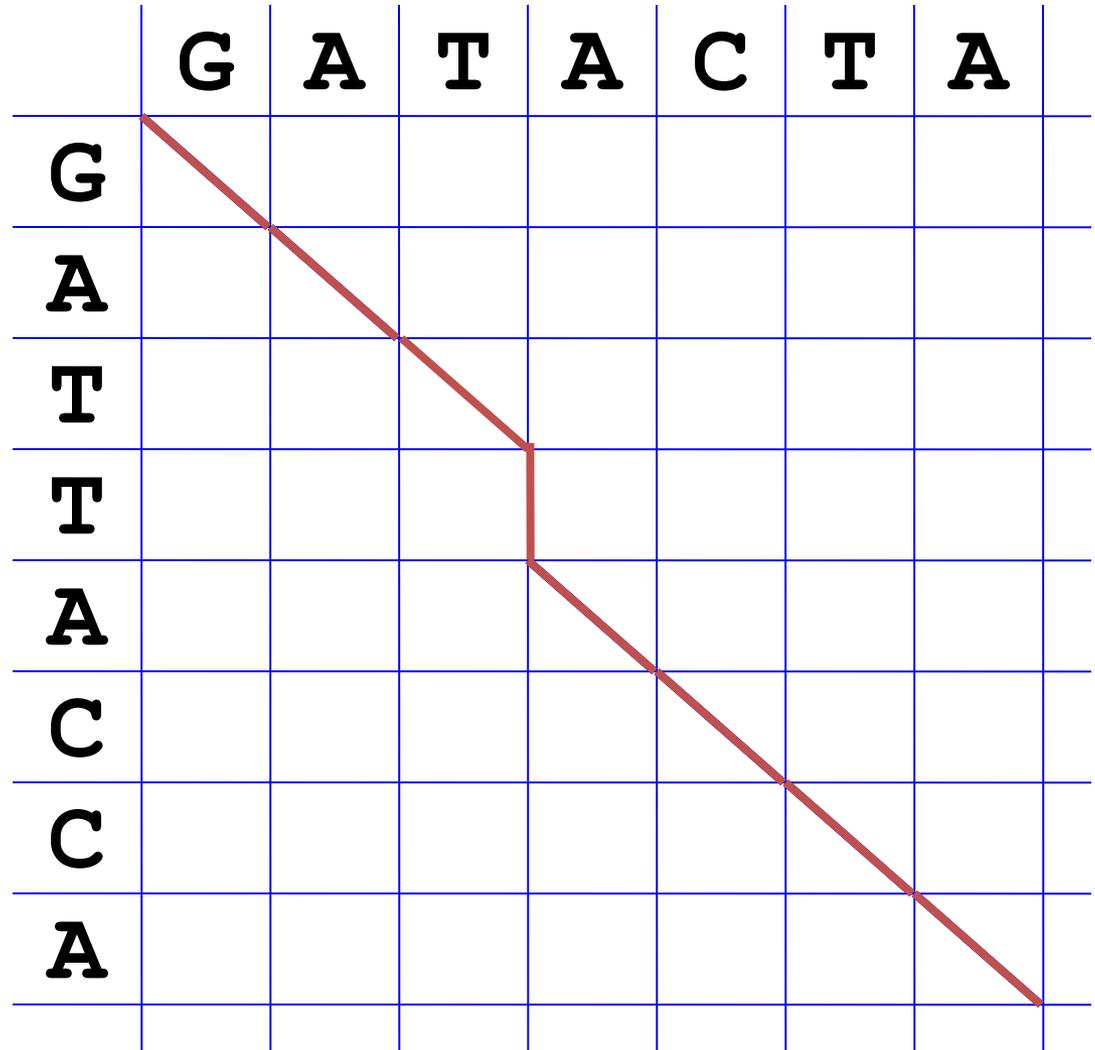
**GA-TACTA**  
**GATTACCA**



# Dynamic programming: alineamiento obtenido

Imprimir el alineamiento

**GAT-ACTA**  
**GATTACCA**



# Dynamic programming: Smith-Waterman

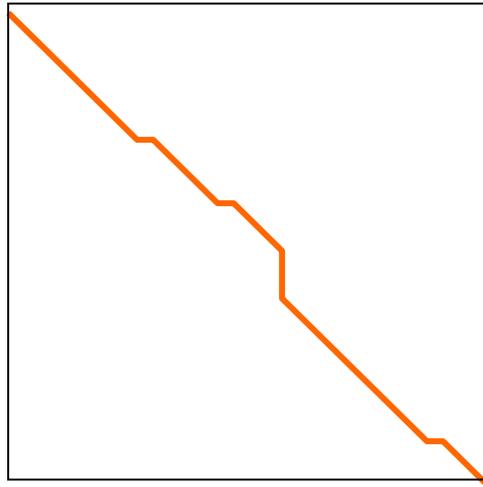
- **El método fue modificado (Smith-Waterman) para obtener alineamientos locales**
- **El método garantiza la obtención de un alineamiento óptimo (cuyo score no puede ser mejorado)**
- **La complejidad es proporcional al producto de las longitudes de las secuencias a alinear**

# Similitud global y local

**El algoritmo de programación dinámica puede ser implementado para alineamientos locales o globales.**

Optimal global alignment

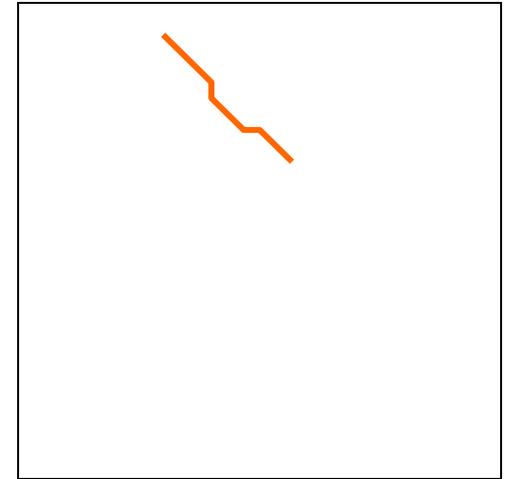
Needleman & Wunsch (1970)



Las secuencias se alinean esencialmente de un extremo a otro

Optimal local alignment

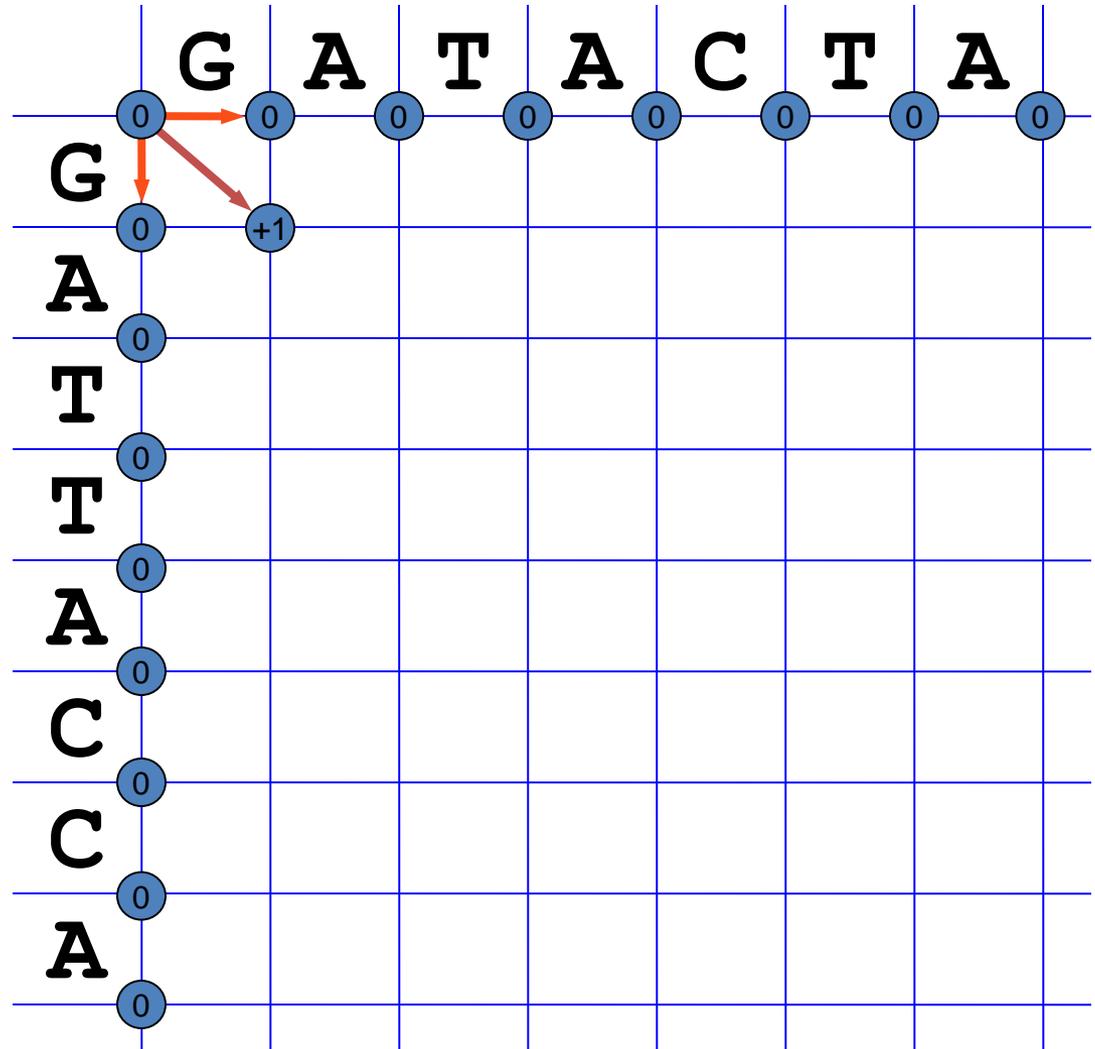
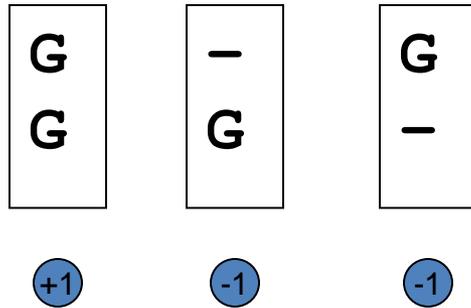
Smith & Waterman (1981)



Las secuencias se alinean en regiones pequeñas y aisladas

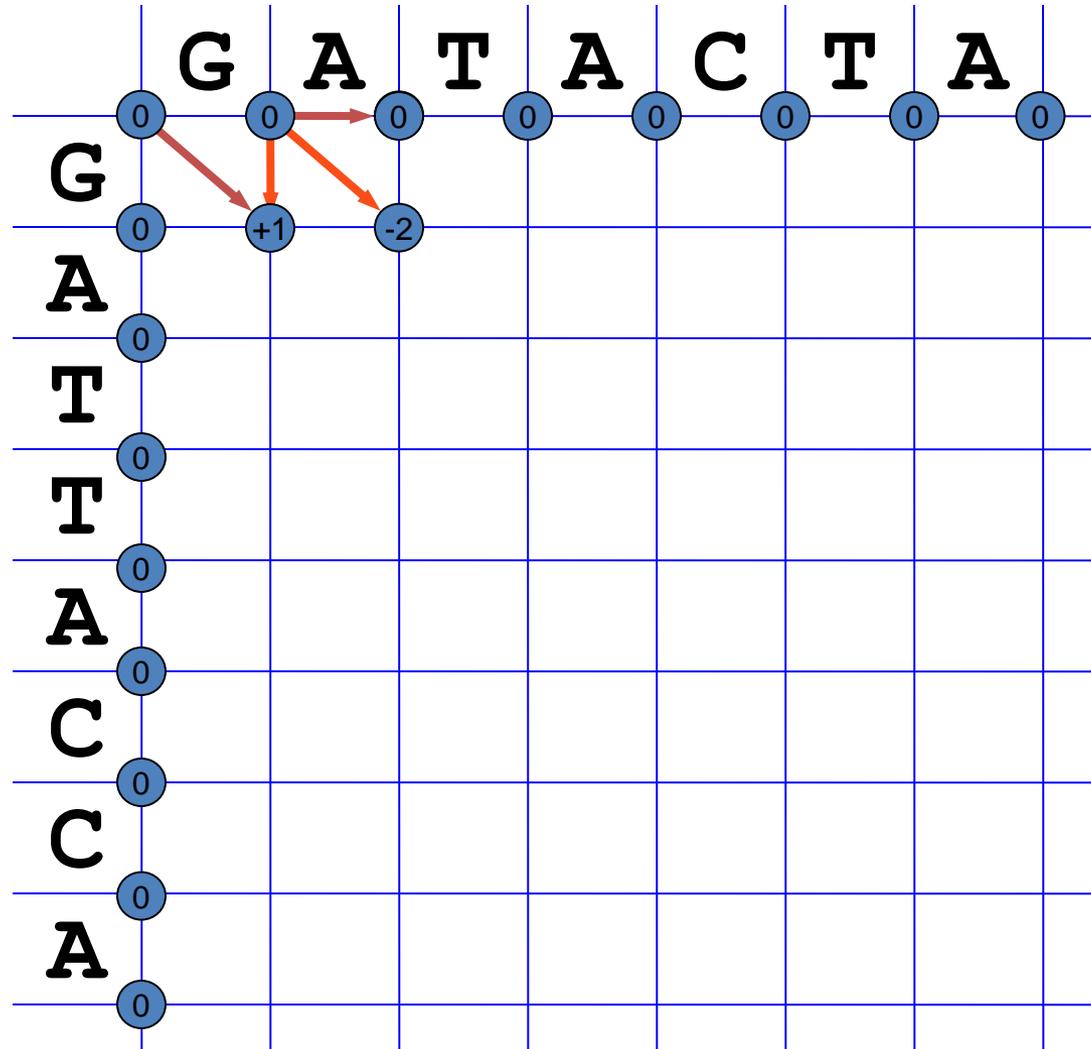
# Smith-Waterman: paso a paso

## Extender el path paso por paso



# Smith-Waterman: paso a paso (2)

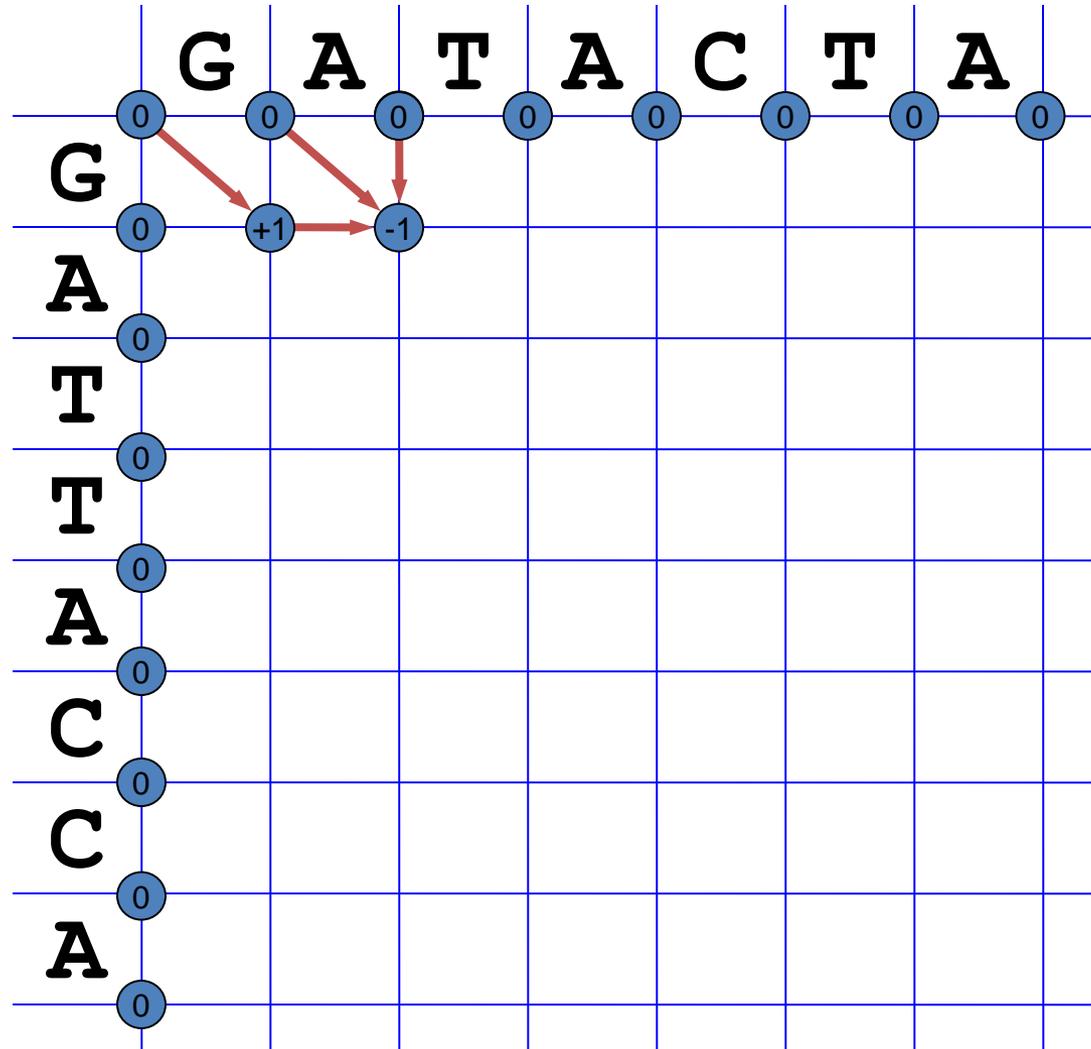
Incrementar el path paso a paso



Recordar el mejor subpath que lleva a cada punto en la matriz.

# Smith-Waterman: paso a paso (3)

Incrementar el path paso a paso



Recordar el mejor subpath que lleva a cada punto en la matriz.

# Dynamic programming demos

- Needleman-Wunsch web app desarrollada para enseñanza

- <http://rna.informatik.uni-freiburg.de/Teaching/index.jsp?toolName=Needleman-Wunsch>

**Input:**

Sequence *a*:

Sequence *b*:

Optimization of: Distance  Similarity

Scoring in *s*: Match  Mismatch  Gap

Hint:  
For similarity maximization,  
match scores should be positive and all other scores lower.  
For distance minimization the reverse applies.

Recursion: 
$$D_{i,j} = \max \begin{cases} D_{i-1,j-1} + s(a_i, b_j) \\ D_{i-1,j} + s(a_i, -) \\ D_{i,j-1} + s(-, b_j) \end{cases} = \max \begin{cases} D_{i-1,j-1} + 1 & a_i = b_j \\ D_{i-1,j-1} + -1 & a_i \neq b_j \\ D_{i-1,j} + -2 & b_j = - \\ D_{i,j-1} + -2 & a_i = - \end{cases}$$

**Output:**

<i>D</i>		A <sub>1</sub>	A <sub>2</sub>	C <sub>3</sub>	G <sub>4</sub>	
		0	-2	-4	-6	-8
A <sub>1</sub>	-2	1	-1	-3	-5	
A <sub>2</sub>	-4	-1	2	0	-2	
T <sub>3</sub>	-6	-3	0	1	-1	
C <sub>4</sub>	-8	-5	-2	1	0	
G <sub>5</sub>	-10	-7	-4	-1	2	

Score: 2

**Results**  
You can select a result to get the related traceback.

AATCG  
\*\* \*\*  
AA\_CG

# Dynamic programming demos

- Smith-Waterman web app desarrollada para enseñanza
  - <http://rna.informatik.uni-freiburg.de/Teaching/index.jsp?toolName=Smith-Waterman>

**Input:**

Sequence *a*:

Sequence *b*:

Scoring in *s*: Match  Mismatch  Gap

Hint:  
For similarity maximization,  
match scores should be positive and all other scores lower.

Recursion: 
$$S_{i,j} = \max \begin{cases} S_{i-1,j-1} + s(a_i, b_j) \\ S_{i-1,j} + s(a_i, -) \\ S_{i,j-1} + s(-, b_j) \\ 0 \end{cases} = \max \begin{cases} S_{i-1,j-1} + 1 & a_i = b_j \\ S_{i-1,j-1} + -1 & a_i \neq b_j \\ S_{i-1,j} + -2 & b_j = - \\ S_{i,j-1} + -2 & a_i = - \\ 0 \end{cases}$$

**Output:**

<i>S</i>		A <sub>1</sub>	A <sub>2</sub>	C <sub>3</sub>	G <sub>4</sub>	
		0	0	0	0	
A <sub>1</sub>		0	1	1	0	
A <sub>2</sub>		0	1	2	0	
T <sub>3</sub>		0	0	0	1	
C <sub>4</sub>		0	0	0	1	
G <sub>5</sub>		0	0	0	0	2

Score: 2

**Results**  
You can select a result to get the related traceback.

AA  
\*\*  
AA  
CG  
\*\*  
CG

- **Un algoritmo de alineamiento local, siempre produce alineamientos locales?**
- **Un algoritmo de alineamiento global siempre produce alineamientos globales?**
- **NO**
  - **dependiendo del sistema de scoring (scores para match/mismatch/gaps) SW puede producir alineamientos globales**
  - **dependiendo la penalidad asignada a los gaps en los extremos de un alineamiento global (o alterando significativamente el sistema de scoring) NW puede producir alineamientos locales**

- **Un sistema de scoring simple, penaliza por igual cualquier mismatch**
- **Biológicamente tiene sentido penalizar ciertos cambios y ser más permisivo con otros**
  - **En proteínas: residuos hidrofóbicos reemplazados entre sí.**
  - **En DNA: transversiones vs transiciones**
- **Una matriz no es otra cosa que un sistema de scoring que permite asignar puntajes individuales a cada una de las letras del alfabeto en uso.**

- **Un ejemplo de matriz de scoring podría ser el clásico ejemplo de penalizar más los cambios que alteran las propiedades químicas de un residuo (aa)**
  - **hidrofóbicos: Ile, Val, Leu, Ala**
  - **Polares (+): Lys, Arg**
  - **Polares (-): Glu, Asp**
  - **Aromáticos: Phe, Tyr, Trp**
  - **etc.**

Ile x Val = -1

Ile x Asp = -5

Phe x Tyr = -1

Phe x Gly = -8

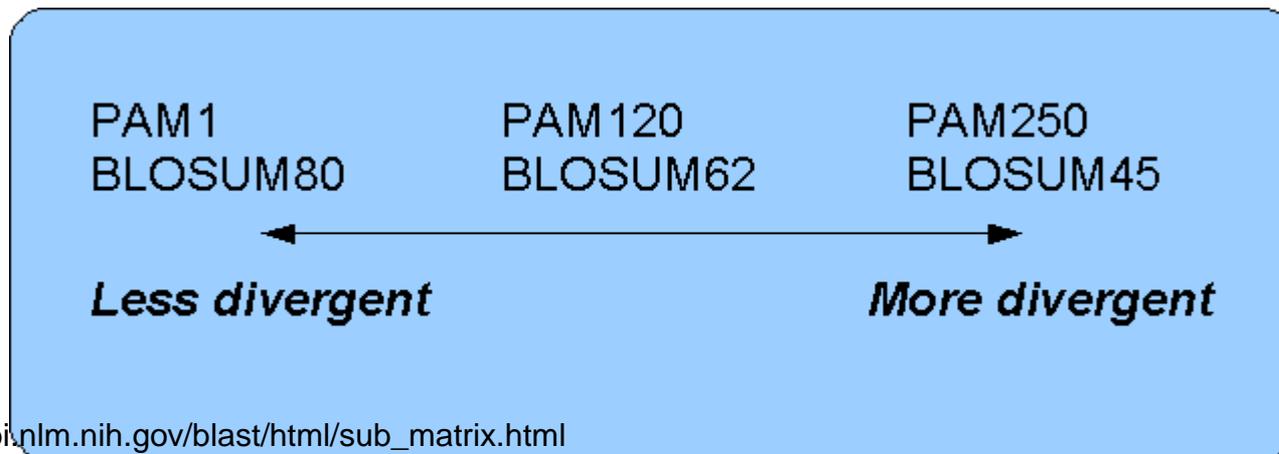
- **PAM (Dayhoff, 1978)**

- **proveen estimaciones de plausibilidad de cambio de un aminoácido en otro en proteínas homólogas**
- **derivadas a partir de un grupo de secuencias > 85% similares**
- **los cambios de aminoácidos observados son llamados “accepted mutations”**
- **Se extrapolan matrices a períodos evolutivos más largos**

# Matrices derivadas por observación

- **BLOSUM (Henikoff)**

- **Blocks Amino Acid Substitution Matrices**
- **Sustituciones de amino ácidos observadas en un conjunto grande de 'blocks'**
- **Representan más de 500 familias de proteínas**
  
- **Se agrupan los blocks de acuerdo a su identidad y se generan matrices**
- **blocks 80% idénticos -> BLOSUM80**
- **Blocks 60% idénticos -> BLOSUM60**
- **etc**



[https://www.ncbi.nlm.nih.gov/blast/html/sub\\_matrix.html](https://www.ncbi.nlm.nih.gov/blast/html/sub_matrix.html)



# Sistemas de scoring: BLOSUM62: identidades

Las identidades tienen scores positivos, pero algunas son más valoradas que otras.

## BLOSUM62

A	4																			
R	-1	5																		
N	-2	0	6																	
D	-2	-2	1	6																
C	0	-3	-3	-3	9															
Q	-1	1	0	0	-3	5														
E	-1	0	0	2	-4	2	5													
G	0	-2	0	-1	-3	-2	-2	6												
H	-2	0	1	-1	-3	0	0	-2	8											
I	-1	-3	-3	-3	-1	-3	-3	-4	-3	4										
L	-1	-2	-3	-4	-1	-2	-3	-4	-3	2	4									
K	-1	2	0	-1	-3	1	1	-2	-1	-3	-2	5								
M	-1	-1	-2	-3	-1	0	-2	-3	-2	1	2	-1	5							
F	-2	-3	-3	-3	-2	-3	-3	-3	-1	0	0	-3	0	6						
P	-1	-2	-2	-1	-3	-1	-1	-2	-2	-3	-3	-1	-2	-4	7					
S	1	-1	1	0	-1	0	0	0	-1	-2	-2	0	-1	-2	-1	4				
T	0	-1	0	-1	-1	-1	-1	-2	-2	-1	-1	-1	-1	-2	-1	1	5			
W	-3	-3	-4	-4	-2	-2	-3	-2	-2	-3	-2	-3	-1	1	-4	-3	-2	11		
Y	-2	-2	-2	-3	-2	-1	-2	-3	2	-1	-1	-2	-1	3	-3	-2	-2	2	7	
V	0	-3	-3	-3	-1	-2	-2	-3	-3	3	1	-2	1	-1	-2	-2	0	-3	-1	4
	A	R	N	D	C	Q	E	G	H	I	L	K	M	F	P	S	T	W	Y	V



# Más matrices

- **PAM**
- **BLOSUM**
- **Otras**
  - **Comparación simple de propiedades químicas de amino ácidos**
  - **Análisis complejos de sustituciones en estructura secundaria de proteínas, a partir de alineamientos estructurales**
  - **Gonnet (1994). Sustitución de dipéptidos**
  - **Jones (1994) matriz específica de proteínas transmembrana**
- **Algunas de estas matrices sirven para alinear proteínas en base a características estructurales y pueden no ser útiles para análisis evolutivos!**

**Bioinformatics. Sequence and Genome analysis. David W Mount, CSHL Press (2001)**

**Introduction to Bioinformatics. Lesk, A. M. (2019). Oxford University Press.**

**Hugues Sicotte (NCBI). (slides DP)**

**Javascript-based implementations for various algorithms  
<http://rna.informatik.uni-freiburg.de/Teaching/>**