CAPITULO IV
ALINEAMIENTO DE SECUENCIAS

Universidad Nacional de San Antonio Abad del Cusco

Departamento Académico de Informática

Dr. Sc. Luis Palma

Alineamiento

Introducción

Comparar secuencias de ADN de dos individuos nos permite determinar:

- Si ambas tienen funcionalidades similares.
- Si ambas tienen funcionalidades parecidas.
- Que hayan evolucionado dependientemente.
- Comparten algún ancestro en común.

La técnica más común de comparar pares de secuencias (nucleótidos, aminoácidos, genes, proteínas) es el **alineamiento**.

Alineamiento de Secuencias

- El alineamiento de secuencias es el **algoritmo más aplicado** en bioinformática
- Su objetivo es **alinear** dos o más secuencias (de ADN o proteínas) de forma que puedan **destacarse las regiones similares** entre las moléculas.
- Al determinar si una secuencia desconocida es similar, en algún sentido, a secuencias conocidas (e idealmente de estructura y función conocidas) **podremos identificarla y predecir su estructura y función**.
- El objetivo del alineamiento es **conseguir alinear las posiciones homólogas**.

Alineamiento de Secuencias

```
CGATGCTAGCGTATCGTAGTCTATCGTAC
                |       ||
ACGATGCTAGCGTTTCGTATCATCGTA
```

Sin Alinear

```
-CGATGCTAGCGTATCGTAGTCTATCGTAC
 |||||||||||| ||||||||||||||||
ACGATGCTAGCGTTTCGTA-TC-ATCGTA-
```

Alineadas

Alineamiento con gap

- Cuando **falta un nucleótido** decimos que hay un **gap**. Un gap puede corresponder a una **deleción** o, a una **inserción**.
- **Pueden existir diferentes alineamientos** dependiendo del **número de gaps** que permitamos introducir.

```
ATATTGCTACGTATATCAT
||||||||||
ATATATGCTACGTATCAT
```

sin gap

```
ATAT-TGCTACGTATATCAT
||||  ||||||||||
ATATATGCTACGTATCAT
```

con gap en una secuencia

```
ATAT-TGCTACGTATATCAT
||||  |||||||   ||||||
ATATATGCTACG--TATCAT
```

con gap en ambas secuencia

Puntuación de Alineamiento

- Para **comparar** distintos **alineamientos** entre sí se pueden **asignar puntuaciones**.
- El alineamiento con **mejor puntuación** debería ser el más razonable desde un punto de vista biológico, el que alinea más posiciones homólogas.
- **Sistemas de puntuación**:
  - Número de letras que coinciden
  - Porcentaje de identidad, número de coincidencias cada cien posiciones.
  - Porcentaje de similitud, tiene en cuenta la similitud fisicoquímica de los diferentes aminoácidos.

Puntuación de Alineamiento

- Se suelen incluir dos **penalizaciones** para los gaps, una para abrir el gap y otra para extenderlo. Este último suele ser menos costoso.
- De entre todos los alineamientos posibles el **óptimo** es el que presenta una **máxima puntuación** para el sistema de puntuación dado.

```
G-ATESLIKESCHEESE
|  |||        ||||||
GRATED-----CHEESE
```

$$\text{Puntuación} = \begin{matrix} 10\ match & \times 1 \\ 1\ mismatch & \times 0 \\ 6\ gaps & \times -1 \end{matrix} = 4$$

Ejemplo: sistema de puntuación: match: +1, mismatch: 0, gap: -1

Alineamiento Global y Local

- Existe dos tipos de alineamientos principales: **globales** y **locales**.
- En el global se intenta que el alineamiento **cubra las dos secuencias completamente introduciendo** lo **gaps** que sean necesarios.
- En el **local se alinean sólo las zonas más parecidas**.
- El **global** sirve para alinear **secuencias que se empiecen y acaben en la misma región**, por ejemplo **genes homólogos** de especies similares.
- El alineamiento **local** sirve para alinear secuencias homólogas **se parecen sólo en las regiones más conservadas**.

Alineamiento Global y Local

```
TAGCTACTCGTAG
|||| ||||
GCTAGTCGT
```

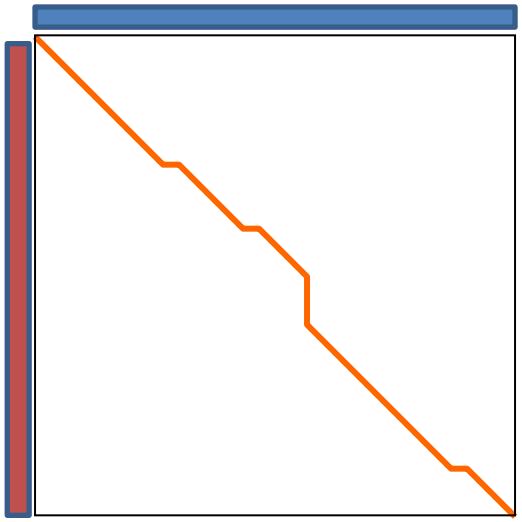
Alineamiento local

```
TACGGGGCTAGCTA-TCGTAG
||||     |||     ||||||
TAGC----TAG----TCGTAG
```

Alineamiento global

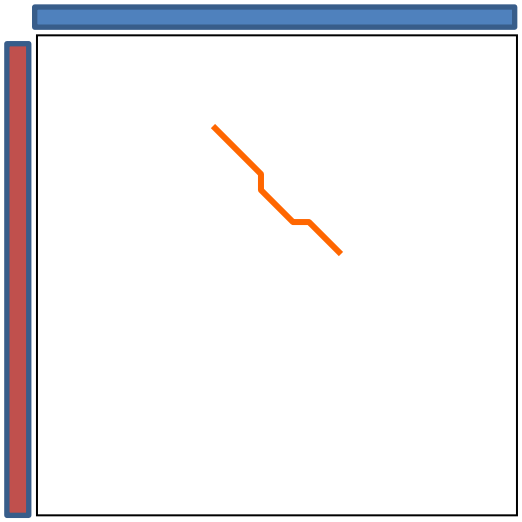
Alineamiento Global y Local

Alineamiento global



Las secuencias se alinean de un extremo a otro

Alineamiento local



Las secuencias se alinean en regiones pequeñas y aisladas

Aplicación de Alineamiento

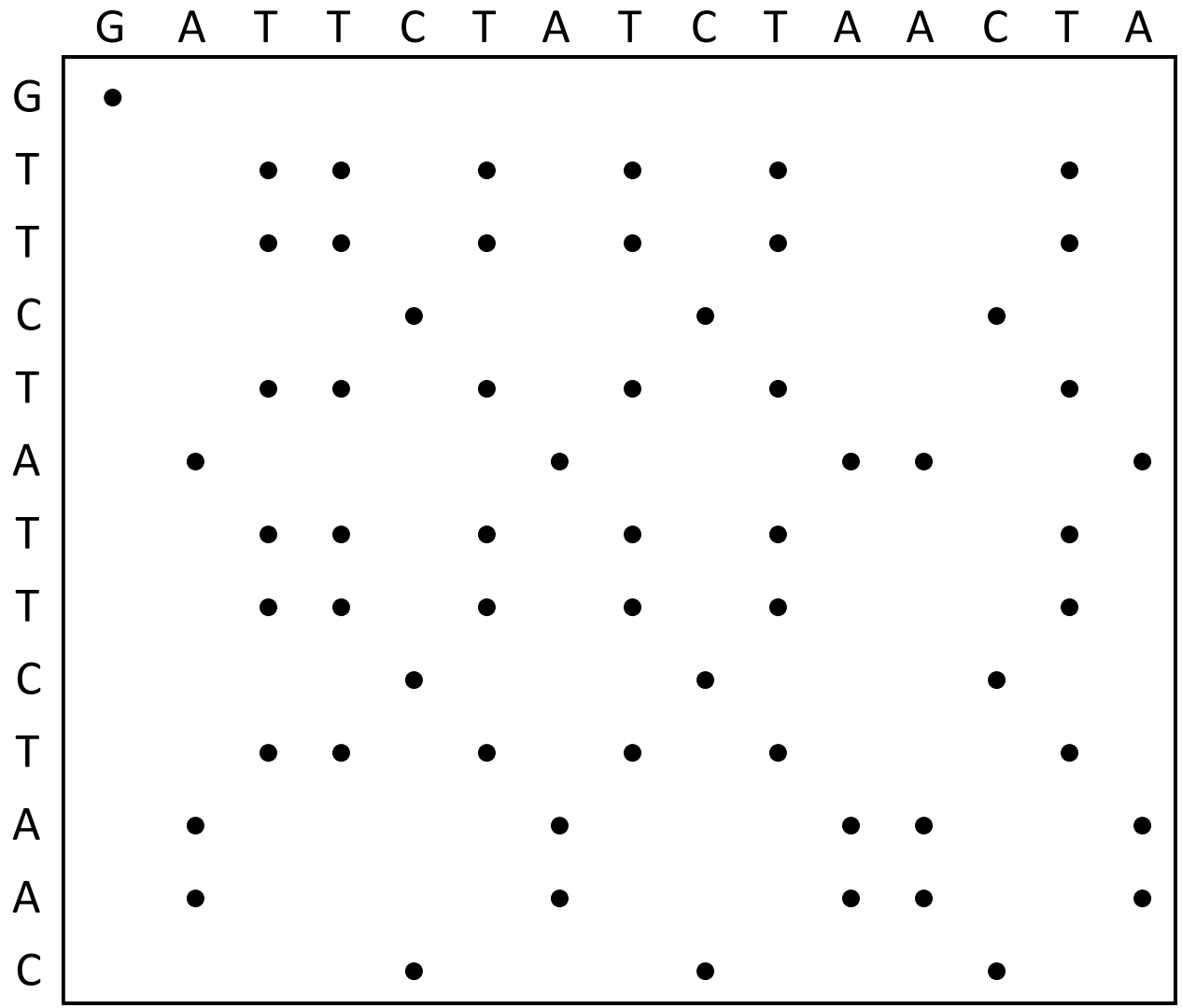
- Mediante un alineamiento **global** entre genomas se puede
  - Identificar repeticiones *internas* (G1 vs G2)
  - Encontrar secuencias conservadas *entre especies* (G1 vs G2)
- Para **predecir la función de una proteína** desconocidas.
  - Mediante alineamientos **locales** entre dos secuencias
  - Mediante alineamientos **múltiples** entre conjuntos de secuencias
- Para **buscar posiciones homólogas** en las secuencias.
- **Comparar un gen y su producto**.

Algoritmo de Alineamiento

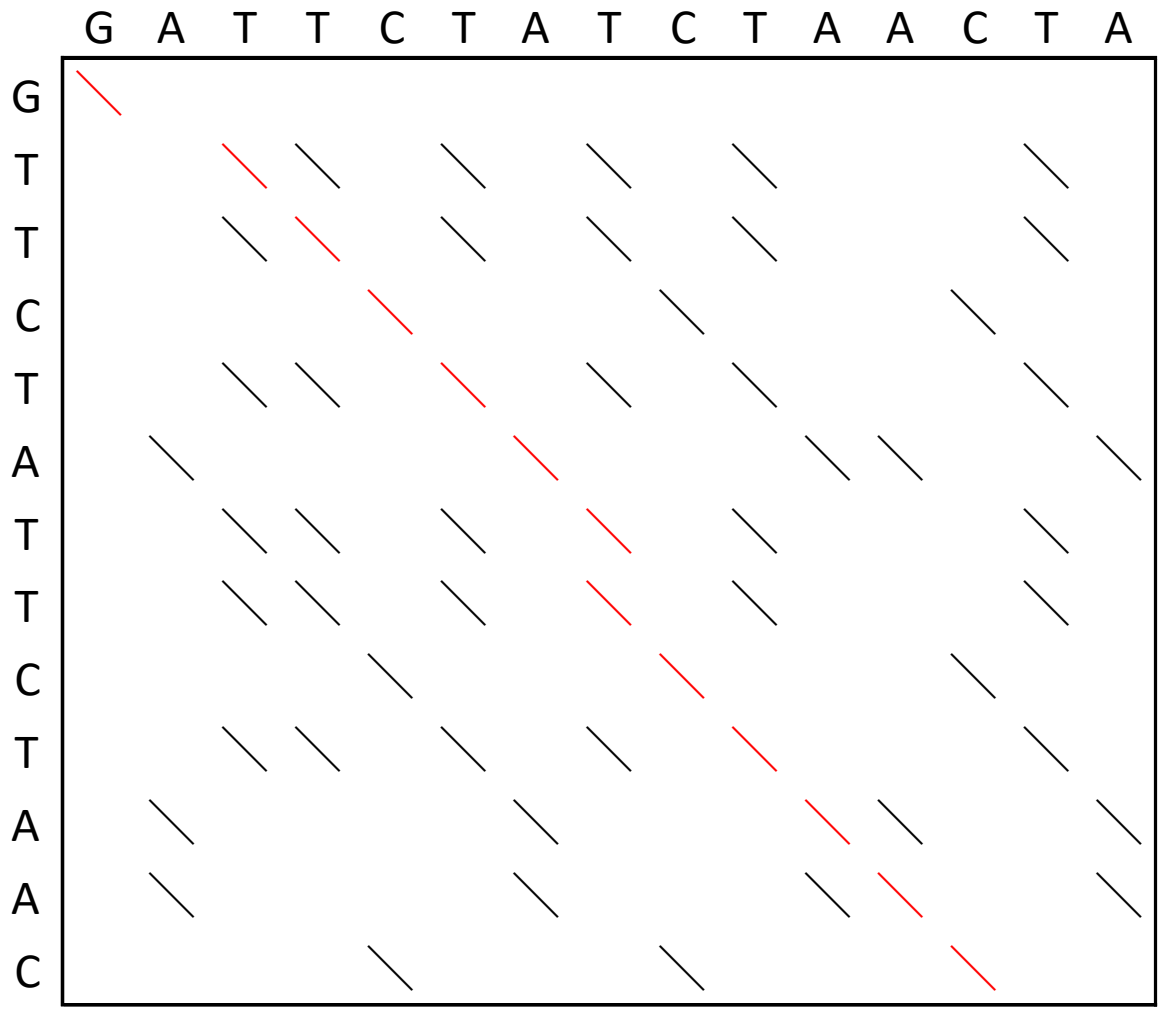
- Matriz de puntos.
  - Dotmatches
- Programación dinámica.
  - Needleman y Wunsch
  - Smith Waterman

Matriz de Puntos

Matriz de puntos



Matriz de puntos



Needleman y Wunsch

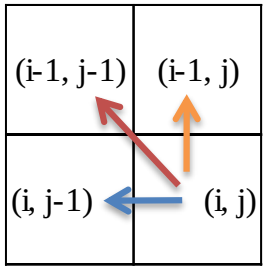
Algoritmo de Needleman y Wunsch

- Paso 1: Colocar cadenas a alinear en la fila y columna de la matriz de puntuación.
- Paso 2: Rellenar primera fila y columna de la matriz de puntuación:

  i*gap_penality, $\forall$ i $\in$ [1…n]

  j*gap_ penality, $\forall$ j $\in$ [1…n]
- Paso 3: Rellenar matriz de puntuación.

$$M_{ij} = Máximo \begin{cases} diagonal + similitud \\ izquierda + gap\_penality \\ arriba + gap\_penality \end{cases}$$



- Paso 4: Realizar el rastreo hacia atrás.

↖	Carácter **Fila** Carácter **Column a**
←	Hueco **Fila** Carácter **Columna**
↑	Carácter **Fila** Hueco **Columna**

- Paso 5: Construir el alineamiento de secuencias.

Ejemplo - Needleman y Wunsch

- Alinear mediante Needleman Wunsch, las siguientes secuencias:

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

T C G A C A C T

- Si la matriz de similitud es:

	A	C	G	T
A	2			
C	-3	2		
G	-3	-3	2	
T	-3	-3	-3	2

- y el costo del gap es: **-2**

Ejemplo - Needleman y Wunsch

- Paso 1: Colocar cadenas a alinear en la fila y columna de la matriz de puntuación.

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

Ejemplo - Needleman y Wunsch

- Paso 2: Rellenar primera fila y columna de la matriz de puntuación:

  $i*gap\_penality, \forall\ i \in [1 \ldots n]$

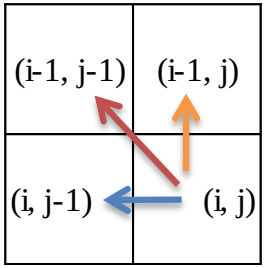
  $j*gap\_\ penality, \forall\ j \in [1 \ldots n]$

		C	A	G	T	C	C	A	A	A	T	A	G	C	T
	0	-2	-4	-6	-8	-10	-12	-14	-16	-18	-20	-22	-24	-26	-28
T	-2														
C	-4														
G	-6														
A	-8														
C	-10														
A	-12														
C	-14														
T	-16														

Ejemplo - Needleman y Wunsch

- Paso 3: Rellenar matriz de puntuación.

$$M_{ij} = Máximo \begin{cases} diagonal + similitud \\ izquierda + gap\_penality \\ arriba + gap\_penality \end{cases}$$



		C	A	G	T	C	C	A	A	A	T	A	G	C	T
	0	-2	-4	-6	-8	-10	-12	-14	-16	-18	-20	-22	-24	-26	-28
T	-2	-3	-5	-7	-4	-6	-8	-10	-12	-14	-16	-18	-20	-22	-24
C	-4	0	-2	-4	-6	-2	-4	-6	-8	-10	-12	-14	-16	-18	-20
G	-6	-2	-3	0	-2	-4	-5	-7	-9	-11	-13	-15	-12	-14	-16
A	-8	-4	0	-2	-3	-5	-7	-3	-5	-7	-9	-11	-13	-15	-17
C	-10	-6	-2	-3	-5	-1	-3	-5	-6	-8	-10	-12	-14	-11	-13
A	-12	-8	-4	-5	-6	-3	-4	-1	-3	-4	-6	-8	-10	-12	-14
C	-14	-10	-6	-7	-8	-4	-1	-3	-4	-6	-7	-9	-11	-8	-10
T	-16	-12	-8	-9	-5	-6	-3	-4	-6	-7	-4	-6	-8	-10	-6

Ejemplo - Needleman y Wunsch

- Paso 4: Realizar el rastreo hacia atrás.

↖	Carácter **Fila** Carácter **Column a**	
←	Hueco **Fila** Carácter **Columna**	
↑	Carácter **Fila** Hueco **Columna**	

		C	A	G	T	C	C	A	A	A	T	A	G	C	T
	0	-2	-4	-6	-8	-10	-12	-14	-16	-18	-20	-22	-24	-26	-28
T	-2	-3	-5	-7	-4	-6	-8	-10	-12	-14	-16	-18	-20	-22	-24
C	-4	0	-2	-4	-6	-2	-4	-6	-8	-10	-12	-14	-16	-18	-20
G	-6	-2	-3	0	-2	-4	-5	-7	-9	-11	-13	-15	-12	-14	-16
A	-8	-4	0	-2	-3	-5	-7	-3	-5	-7	-9	-11	-13	-15	-17
C	-10	-6	-2	-3	-5	-1	-3	-5	-6	-8	-10	-12	-14	-11	-13
A	-12	-8	-4	-5	-6	-3	-4	-1	-3	-4	-6	-8	-10	-12	-14
C	-14	-10	-6	-7	-8	-4	-1	-3	-4	-6	-7	-9	-11	-8	-10
T	-16	-12	-8	-9	-5	-6	-3	-4	-6	-7	-4	-6	-8	-10	-6

Ejemplo - Needleman y Wunsch

- Paso 5: Construir el alineamiento de secuencias.

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

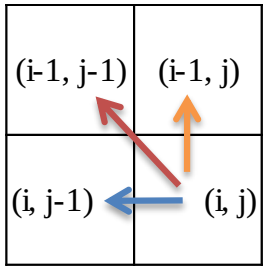
	Cant	Puntaje	Total
Similitud	6	2	12
Dif. A-C	0	-3	0
Dif. A-G	1	-3	-3
Dif. A-T	0	-3	0
Dif. C-G	0	-3	0
Dif. C-T	1	-3	-3
Dif. G-T	0	-3	0
gap	6	-2	-12
			-6

Smith Waterman

Algoritmo de Smith Waterman

- Paso 1: Colocar cadenas a alinear en la fila y columna de la matriz de puntuación.
- Paso 2: Rellenar primera fila y columna de la matriz con CEROS
- Paso 3: Rellenar matriz de puntuación.

$$M_{ij} = Máximo \begin{cases} diagonal + similitud \\ izquierda + gap\_penality \\ arriba + gap\_penality \\ 0 \end{cases}$$



- Paso 4: Realizar el rastreo hacia atrás (valor más alto de la tabla, hasta antes de llegar a un cero).

↖	Carácter **Fila** Carácter **Column a**
←	Hueco **Fila** Carácter **Columna**
↑	Carácter **Fila** Hueco **Columna**

- Paso 5: Construir el alineamiento de secuencias.

Ejemplo – Smith Waterman

- Alinear mediante Smith Waterman, las siguientes secuencias:

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

T C G A C A C T

- Si la matriz de similitud es:

	A	C	G	T
A	2			
C	-3	2		
G	-3	-3	2	
T	-3	-3	-3	2

- y el costo del gap es: **-2**

Algoritmo de Smith Waterman

- Paso 1: Colocar cadenas a alinear en la fila y columna de la matriz de puntuación.

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

Algoritmo de Smith Waterman

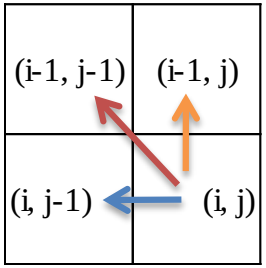
- Paso 2: Rellenar primera fila y columna de la matriz con CEROS

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

Algoritmo de Smith Waterman

- Paso 3: Rellenar matriz de puntuación.

$$M_{ij} = Máximo \begin{cases} diagonal + similitud \\ izquierda + gap\_penality \\ arriba + gap\_penality \\ 0 \end{cases}$$



		C	A	G	T	C	C	A	A	A	T	A	G	C	T
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
T	0	0	0	0	2	0	0	0	0	0	2	0	0	0	2
C	0	2	0	0	0	4	2	0	0	0	0	0	0	2	0
G	0	0	0	2	0	2	1	0	0	0	0	0	2	0	0
A	0	0	2	0	0	0	0	3	2	2	0	2	0	0	0
C	0	2	0	0	0	2	2	1	0	0	0	0	0	2	0
A	0	0	4	2	0	0	0	4	3	2	0	2	0	0	0
C	0	2	2	1	0	2	2	2	1	0	0	0	0	2	0
T	0	0	0	0	3	1	0	0	0	0	2	0	0	0	4

Algoritmo de Smith Waterman

- Paso 4: Realizar el rastreo hacia atrás (valor más alto de la tabla, hasta antes de llegar a un cero).

		C	A	G	T	C	C	A	A	A	T	A	G	C	T
	0	←	←	←	←	←	←	←	←	←	←	←	←	←	←
T	↑	←	←	←	↖	←	←	←	←	←	↖	←	←	←	↖
C	↑	↖	←	←	↑	↖	↖	←	←	←	↑	←	←	↖	↑
G	↑	↑	←	↖	←	↑	↖	←	←	←	←	←	↖	↑	←
A	↑	←	↖	↑	←	↑	←	↖	↖	↖	←	↖	↑	←	←
C	↑	↖	↑	←	←	↖	↖	↑	↖	↑	←	↑	←	↖	←
A	↑	↑	↖	←	←	↑	↑	↖	↖	↖	←	↖	←	↑	←
C	↑	↖	↑	↖	←	↖	↖	↑	↖	↖	←	↑	←	↖	←
T	↑	↑	↑	←	↖	←	↑	↑	←	←	↖	←	←	↑	↖

Algoritmo de Smith Waterman

- Paso 5: Construir el alineamiento de secuencias.

C T
| |
C T

	Cant	Puntaje	Total
Similitud	2	2	4
Dif. A-C	0	-3	0
Dif. A-G	0	-3	0
Dif. A-T	0	-3	0
Dif. C-G	0	-3	0
Dif. C-T	0	-3	0
Dif. G-T	0	-3	0
gap	0	-2	0
			4

EJERCICIOS
Needleman Wunsch y Smith Waterman

Ejercicio 01

- Alinear mediante Needleman Wunsch y Smith Waterman, las siguientes secuencias:

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

- Si la matriz de similitud es:

	A	C	G	T
A	3			
C	-1	3		
G	1	-1	3	
T	-1	1	-1	3

- y el costo de Hueco es: **-1**

Ejercicio 02

- Alinear mediante Needleman Wunsch y Smith Waterman, las siguientes secuencias:

TACGGGGCTAGCTATCGTAG

TAGCTAGTCGTAG

- Si la matriz de similitud es:

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

- y el costo de Hueco es: **-1**

Ejercicio 03

- Alinear mediante el algoritmo de Needleman y Wunsch y el algoritmo Smith Waterman, las siguientes secuencias:

GTCCGACTAGTG

CATCGGAGCTG

- Si la matriz de similitud es:

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

- y el costo de Hueco es: **-2**

Ejercicio 04

- Alinear mediante el algoritmo de Needleman Wunsch y Smith Waterman, las siguientes secuencias:

ATCGA

CATAC

- Si la matriz de similitud es:

	A	C	G	T
A	3			
C	-1	3		
G	1	-1	3	
T	-1	1	-1	3

- y el costo de Hueco es: **-2**

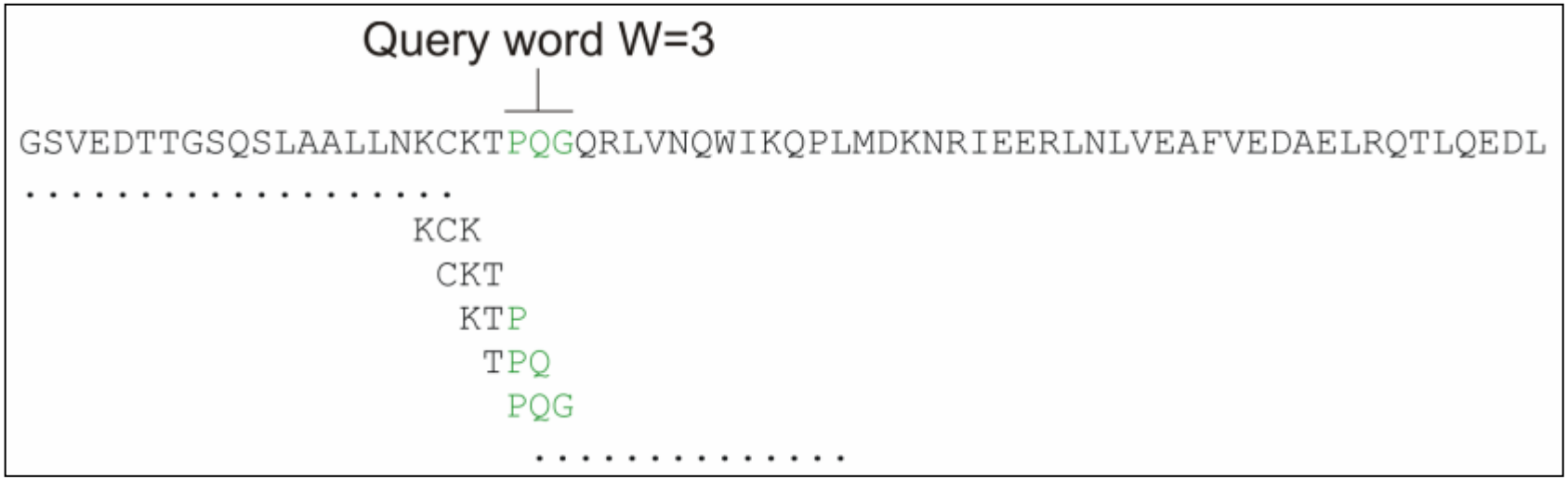
Alineamiento Múltiple

BLAST

- **B**asic **L**ocal **A**lignment **S**earch **T**ool
- Realiza **búsqueda en patrones cortos** más significativos, (tamaño de palabra: 3 proteínas/ 11 nucleótidos).
- Realiza **alineamiento local**.
- Su uso es para **búsqueda en bases de datos**.

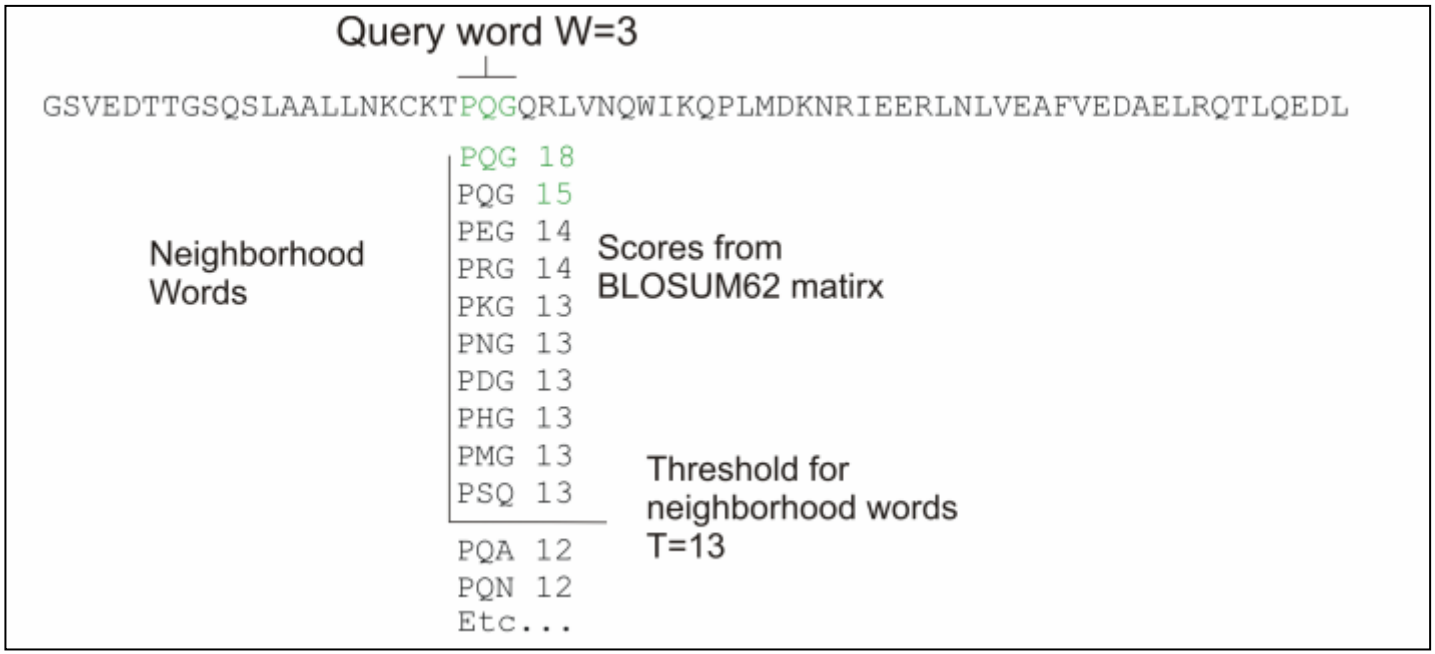
Etapa 1: Procesamiento previo de la secuencia problema.

BLAST intenta encontrar rápidamente las regiones similares entre la secuencia problema y cada secuencia de la BD para no perder tiempo explorando regiones que no guardan ningún parecido. Para ello **divide la secuencia problema en "palabras"** (*words*) con un número de caracteres determinado (*w*). Por ejemplo, a partir de la secuencia RGDVI se obtienen 3 palabras con un tamaño *w* = 3: RGD, GDV y DVI. En el caso de proteínas se suele utilizar un tamaño *w* = 3 y en el caso de DNA se suele utilizar un tamaño *w* = 11.



Etapa 1: Procesamiento previo de la secuencia problema.

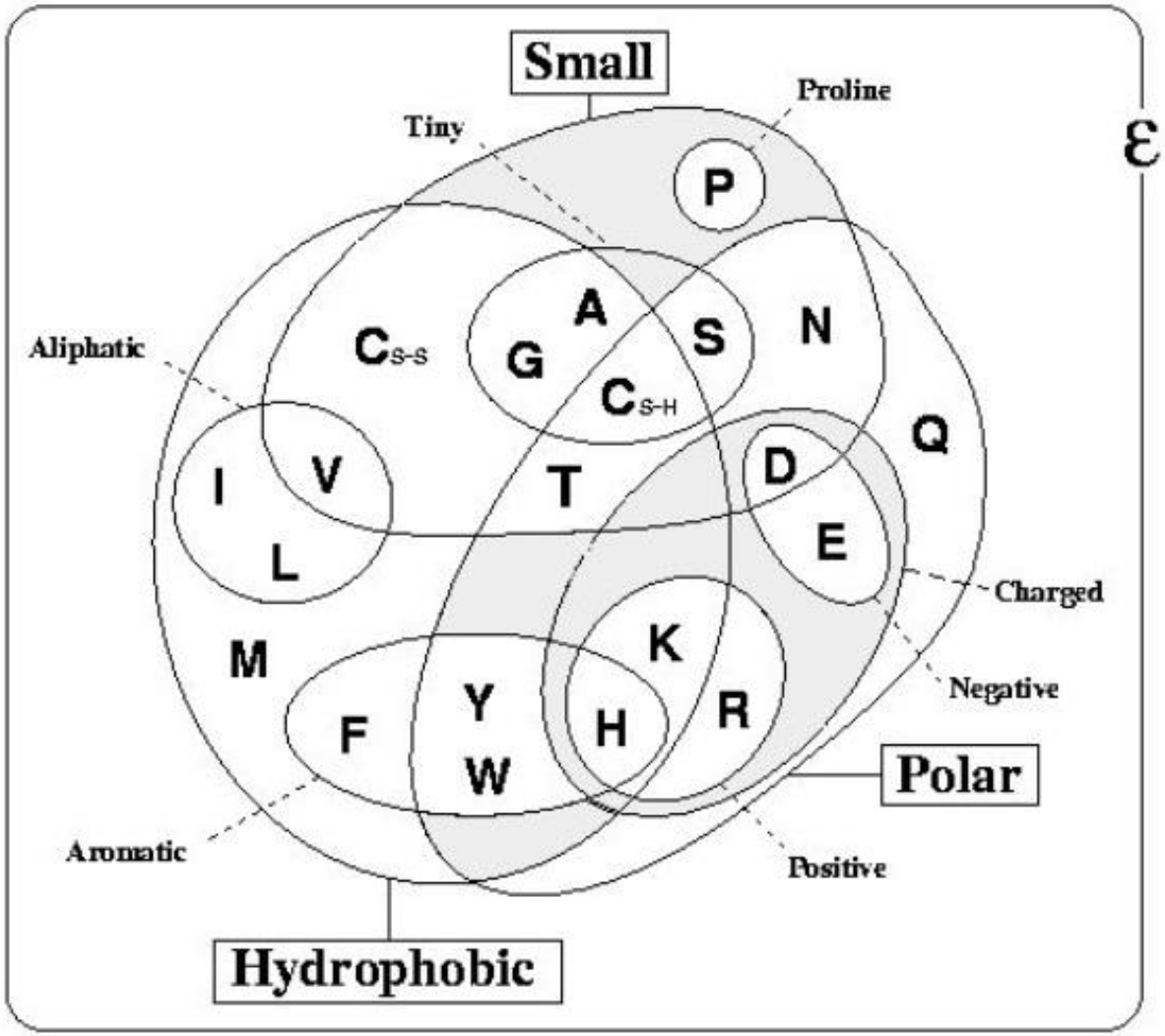
Se supone que los alineamientos significativos deben contener "palabras" idénticas o muy parecidas. Con cada palabra de la secuencia problema se **genera una lista de palabras "parecidas"** **(***neighbors*) que incluye aquellas palabras que, al compararlas con la palabra original de la secuencia problema obtengan una puntuación superior a un valor **T (*threshold*)** utilizando una matriz de puntuación adecuada (por defecto, se utiliza la matriz BLOSUM62).



BLOSUM62: Matriz de sustitución

		A	R	N	D	C	Q	E	G	H	I	L	K	M	F	P	S	T	W	Y	V
Ala	A	4																			
Arg	R	-1	5																		
Asn	N	-2	0	6																	
Asp	D	-2	-2	1	6																
Cys	C	0	-3	-3	-3	9															
Gln	Q	-1	1	0	0	-3	5														
Glu	E	-1	0	0	2	-4	2	5													
Gly	G	0	-2	0	-1	-3	-2	-2	6												
His	H	-2	0	1	-1	-3	0	0	-2	8											
Ile	I	-1	-3	-3	-3	-1	-3	-3	-4	-3	4										
Leu	L	-1	-2	-3	-4	-1	-2	-3	-4	-3	2	4									
Lys	K	-1	2	0	-1	-3	1	1	-2	-1	-3	-2	5								
Met	M	-1	-1	-2	-3	-1	0	-2	-3	-2	1	2	-1	5							
Phe	F	-2	-3	-3	-3	-2	-3	-3	-3	-1	0	0	-3	0	6						
Pro	P	-1	-2	-2	-1	-3	-1	-1	-2	-2	-3	-3	-1	-2	-4	7					
Ser	S	1	-1	1	0	-1	0	0	0	-1	-2	-2	0	-1	-2	-1	4				
Thr	T	0	-1	0	-1	-1	-1	-1	-2	-2	-1	-1	-1	-1	-2	-1	1	5			
Trp	W	-3	-3	-4	-4	-2	-2	-3	-2	-2	-3	-2	-3	-1	1	-4	-3	-2	11		
Tyr	Y	-2	-2	-2	-3	-2	-1	-2	-3	2	-1	-1	-2	-1	3	-3	-2	-2	2	7	
Val	V	0	-3	-3	-3	-1	-2	-2	-3	-3	3	1	-2	1	-1	-2	-2	0	-3	-1	4

Agrupamiento de aminoácidos por características químico físicas

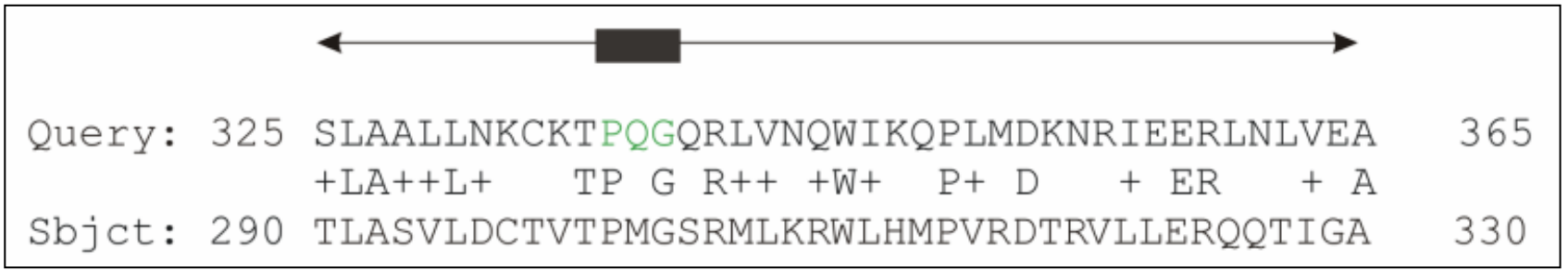


Etapa 1: Procesamiento previo de la secuencia problema.

Ajustando los parámetros T y *w* se puede escoger entre hacer un alineamiento sensible pero lento, o uno más rápido pero con menor sensibilidad. Al disminuir *w* o T aumenta la sensibilidad de la búsqueda (disminuye el número de falsos negativos) pero ésta se hace más lenta.

Etapa 2: Búsqueda de las palabras de las listas en las secuencias de las BD.

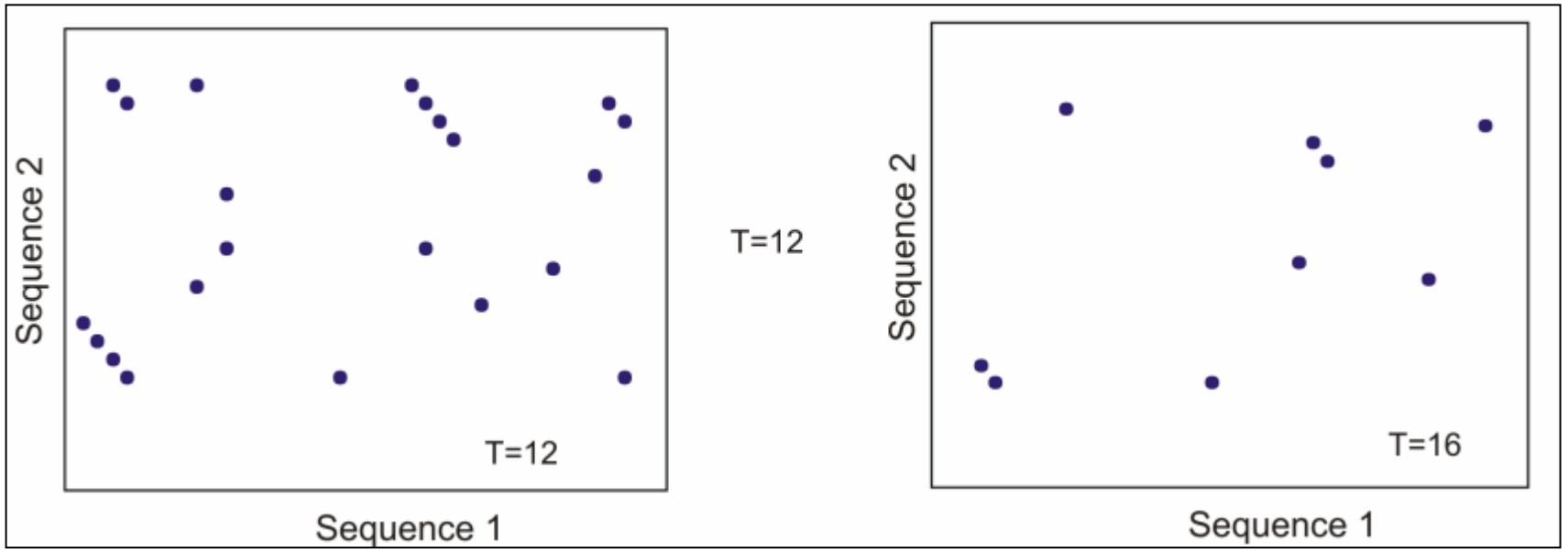
Se buscan las palabras que aparecen en las listas generadas en la etapa anterior en las secuencias de las BD. **Cada vez que se encuentra una coincidencia, se registra su posición** en la memoria del ordenador.



Etapa 2: Búsqueda de las palabras de las listas en las secuencias de las BD.

A partir de este momento, la búsqueda se limita a aquellas regiones en las que se han encontrado coincidencias. Esta es la clave de la rapidez del algoritmo, ya que **se reduce extraordinariamente el espacio de búsqueda**. Si aumentamos el valor de T se obtienen menos coincidencias, se reduce todavía más el espacio de búsqueda y BLAST funciona con mayor rapidez, pero aumentan las probabilidades de pasar por alto algún alineamiento significativo (aumentan los falsos negativos) y se reduce la sensibilidad.

Etapa 2: Búsqueda de las palabras de las listas en las secuencias de las BD.

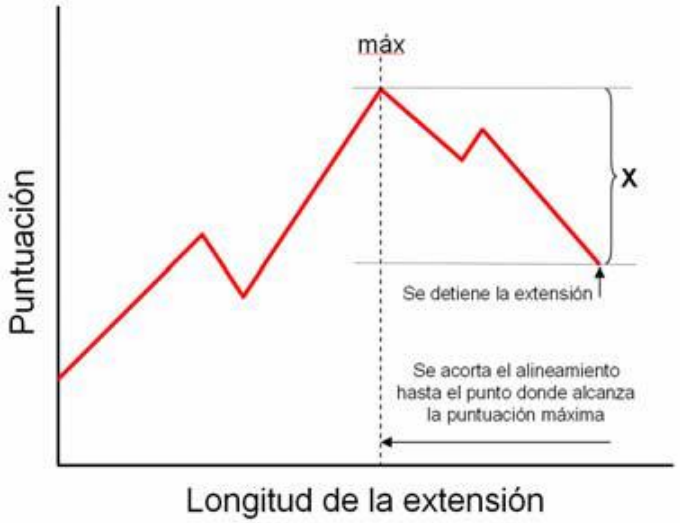


Para hacer que los alineamientos encontrados sean significativos **se suelen enmascarar las regiones de la secuencia que presentan baja complejidad**. Estas regiones contienen secuencias repetidas que pueden ofrecer alineamientos de escaso o nulo interés biológico.

Etapa 3: Extensión

En esta etapa BLAST intenta **extender el alineamiento a ambos lados de cada coincidencia sin dejar huecos**, utilizando el algoritmo de Smith-Waterman. Así se podrá determinar si la coincidencia forma parte de una subsecuencia más larga donde existe similitud entre las dos secuencias. En este caso, la puntuación del alineamiento local aumenta a medida que éste se va extendiendo en las dos direcciones.

For each word match («hit»), extend ungapped alignment in both directions. Stop when S decreases by more than X from the highest value reached by S.



Etapa 3: Extensión

El proceso de extensión se detiene en el momento en que la puntuación acumulada alcanza un valor igual a la puntuación máxima registrada menos un valor X, que es un parámetro del programa. Cuando se detiene la extensión, **el alineamiento retrocede** hasta el punto en que alcanzó la puntuación máxima.

El programa selecciona aquellos alineamientos que tienen una puntuación igual o mayor que ***S* (*score*),** uno de los parámetros ajustables del programa. Cada alineamiento que cumple con esta condición recibe el nombre de **HSP (*High-scoring Segment Pair*)** (Par de segmentos de alta puntuación) y el que obtiene la máxima puntuación es el **MSP (*Maximal Segment Pair*)** (Par de segmento máximo).

Etapa 4: Evaluación

Nunca hay que extraer conclusiones sobre el significado biológico de un alineamiento basándose exclusivamente en su puntuación. Lo primero que hay que hacer es **determinar es si el resultado es estadísticamente significativo** y, después, hay que tener en cuenta el contexto biológico de la búsqueda y toda la información que se pueda conseguir por otras vías.

Etapa 4: Evaluación

La significación estadística es una medida de la probabilidad de obtener un alineamiento con esa misma puntuación por simple azar. A partir de la puntuación obtenida (*S*) BLAST **calcula la significación estadística** de cada alineamiento. Como las puntuaciones de los MSP se ajustan a una **distribución de valores extremos**, es posible calcular la probabilidad de que un MSP obtenga una puntuación igual o mayor que *S* por simple casualidad. Esta probabilidad se llama **valor p** y se calcula mediante la siguiente expresión:

$$p(score \geq S) = 1 - exp\left(-K\, m\, n\, e^{-\lambda S}\right)$$

Etapa 4: Evaluación

$$p(score \geq S) = 1 - exp\,(-K\,m\,n\,e^{-\lambda S})$$

donde m y n son las longitudes de las secuencias comparadas, y K (factor de escala) y λ (factor de bajada) son dos parámetros que dependen de la matriz de sustitución empleada.

Cuanto menor sea el valor P, menos probable es que el alineamiento se deba a simple azar. A modo de orientación,

K y λ

Scoring matrix	Gap opening penalty[b]	Gap extension penalty[b]	*K*	λ
BLOSUM50	∞[a]	0-∞	0.232	0.11
BLOSUM50	15	8–15	0.09	0.222
BLOSUM50	11	8–11	0.05	0.197
BLOSUM50	11	1	—	—
BLOSUM62	∞[a]	0-∞	0.318	0.13
BLOSUM62	12	3–12	0.1	0.305
BLOSUM62	8	7–8	0.06	0.270
BLOSUM62	7	1	—	—
PAM250	∞[a]	0-∞	0.229	0.09
PAM250	15	5–15	0.06	0.215
PAM250	10	8-10	0.031	0.175
PAM250	11	1	—	—

Etapa 4: Evaluación

- si **p** < 10^{-100} las secuencias son idénticas y se observarán alineamientos largos que abarcan prácticamente la totalidad de la secuencia.
- si 10^{-100} < **p** < 10^{-50}, se trata de secuencias casi idénticas (alelos o SNP).
- si 10^{-50} < **p** < 10^{-10}, se trata de secuencias estrechamente relacionadas que, probablemente, son homólogas o comparten algún dominio conservado.
- si 10^{-10} < **p** < 10^{-1}, podría haber un parecido remoto.
- si **p** > 10^{-1}, lo más probable es que el parecido no sea significativo.

Etapa 4: Evaluación

Cuando se hace una búsqueda en una BD, además del valor p también se suele incluir el **valor E**: el número de alineamientos con una puntuación igual o mayor que *S* que se espera encontrar por simple azar en una BD de igual tamaño y composición:

$$E = K\, m\, n\, e^{-\lambda S}$$

donde *m* y *n* indican la longitud de la secuencia problema y de la BD, respectivamente, *S* es la puntuación del alineamiento y *K* e λ son dos constantes que dependen del sistema de puntuación.

Etapa 4: Evaluación

Cuanto menor sea el valor E, menos probable es que el alineamiento se deba a una simple casualidad. A modo de orientación,

- si $E \leq 0.02$, es probable que las secuencias sean homólogas.
- si $0.02 < E < 1$, no se puede descartas que sean homólogas.
- si $E \geq 1$, lo más probable es que el alineamiento se deba a una simple casualidad.

Comprensión fácil

E-value (E)

¿Probabilidad de que el Query sea parecido al de la BD por suerte?

Se busca una secuencia de ADN (Query) en la BD y BLAST encuentra algo parecido.

Si E = 0.00001 (cercano a 0)
hay un 0.001% de que el parecido sea casualidad (El parecido es verdad)
Si E = 2 (valor alto)
hay un 200% de que el parecido es casualidad (El parecido no es verdad)

P-value (p)

¿Probabilidad de que el Query exista en al BD por suerte?

Se busca una secuencia de ADN (Query) en la BD, p-value representa la probabilidad al azar de que la secuena aparezca en la BD.

Si p = 0.0001
hay un 0.01% que haya sido suerte que exista parecidos en la BD (buena coincidencia)
Pero si p = 0.5
hay un 50% que haya sido suerte que exista en la BD (no es confiable)

E-value (E)

¿Frecuencia esperada de encontrar un alineamiento tan bueno como este por azar en la base de datos?

Se busca una secuencia de ADN (Query) en la base de datos, y BLAST encuentra algo parecido. El E-value indica **cuántas veces** se esperaría encontrar ese grado de parecido solo **por casualidad**, dadas la longitud de la secuencia y el tamaño de la base de datos.

- Si **E = 0.00001** (cercano a 0):
  ➤ Se espera que ocurra **0.00001 veces por azar** → el alineamiento **es muy significativo** (no es casualidad).
- Si **E = 2** (valor alto):
  ➤ Se espera que ocurran **2 coincidencias similares por azar** → **es probable que el resultado no sea confiable** (podría ser una coincidencia aleatoria).

p-value (p)

¿Probabilidad de que el alineamiento haya ocurrido por azar (hipótesis nula)?

El p-value representa la **probabilidad estadística** de que el alineamiento observado ocurra solo por casualidad. Es una medida clásica de significancia estadística.

- Si **p = 0.0001**:
  ➤ Hay una probabilidad del **0.01%** de que el resultado sea por azar → **es una coincidencia confiable** (muy significativa).
- Si **p = 0.5**:
  ➤ Hay una probabilidad del **50%** de que el resultado sea por azar → **el resultado no es confiable** (puede deberse al azar).

Otras Técnicas de Alineamientos Múltiple

- MegaBLAST
- PHI-BLAST
- RPS-BLAST
- PSI-BLAST
- FASTA
- Clustal
- MUSCLE: Multiple Sequence Comparison by Log-Expectation.
- T-Coffee: Tree-based Consistency Objective Function For alignmEnt Evaluation.
- Benchmarking

TAREA OBLIGATORIA

- Se tiene los siguientes software: BLASTN, BLASTP, BLASTX, TBLASTN, TBLASTX, FAST AII, Dotmatcher, Dottup, Dothelix, MatrixPlot, EMBOSS NEEDLE, EMBOSS WATER, BioPython.
  1. Seleccione 2 herramientas de la relación (colores diferentes).
  2. Describa el propósito de la herramienta.
  3. Haga uso práctico del mismo.
  4. Presente un pequeño informe de la tarea.