Comparación de Secuencias

Motivación

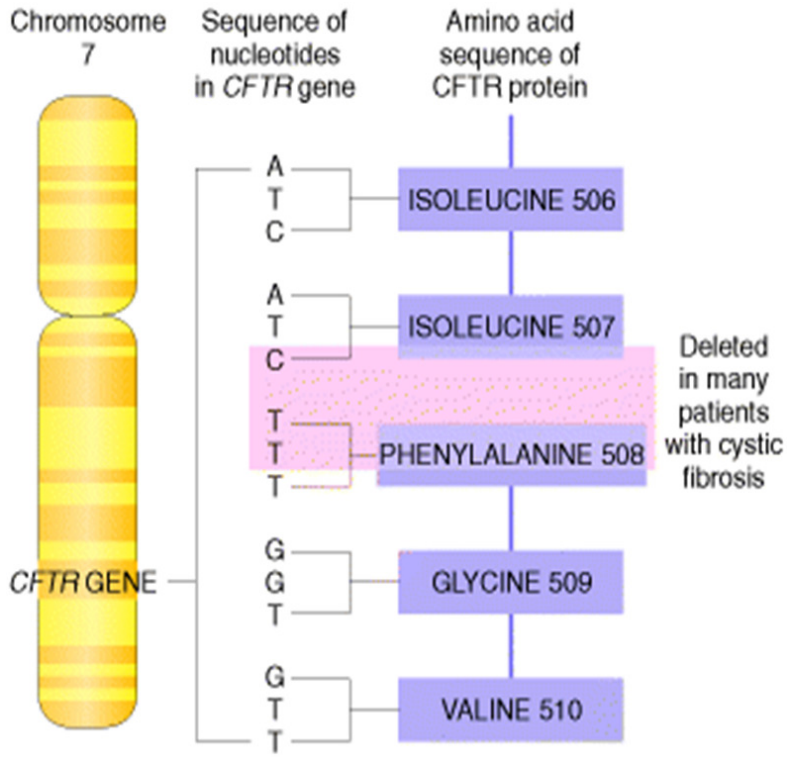
- ¿Por qué es importante realizar la comparación de secuencias?
  - Comparar genomas
  - Buscar genes
  - Buscar secuencias (inicio de transcripción, promotores, exones, intrones)
  - Buscar secuencias similares con genes de función conocida
- Ejemplo
  - Russell Doolittle et al, 1984: encontraron similitudes entre el "recientemente" descubierto *v-sis oncogene*.
  - Se obtuvo un matching entre ese gen y una proteína llamada PDFG (*platelet-derived growth factor*)
  - Cancer podría estar relacionado con un gen normal, haciendo su trabajo bien, pero en el tiempo incorrecto

Motivación

- Fibrosis Cística (CF)
  - Enfermedad genética crónica y normalmente fatal
  - Formación de mucosidad anormalmente espesa y pegajosa en las glándulas
  - Afecta principalmente el sistema respiratorio de los niños
  - Puede afectar también los intestinos, el hígado, las glándulas sudoríparas y los órganos reproductores
  - En 1989 los biólogos encontraron una alta similaridad entre el gen de CF y las ATP binding proteins
  - ATP binding proteins: actuan como canal de transporte en las membranas de la célula
  - Esta relación tiene sentido, dado que CF implica la secreción de sudor con un alto (anormal) contenido the sodio

Motivación

- Fibrosis Cística (CF)
  - En un alto porcentaje (70%)de pacientes con CF se encontró una mutación en el gen CFTR (Cystic Fibrosis Transmebrane conductance regulator)



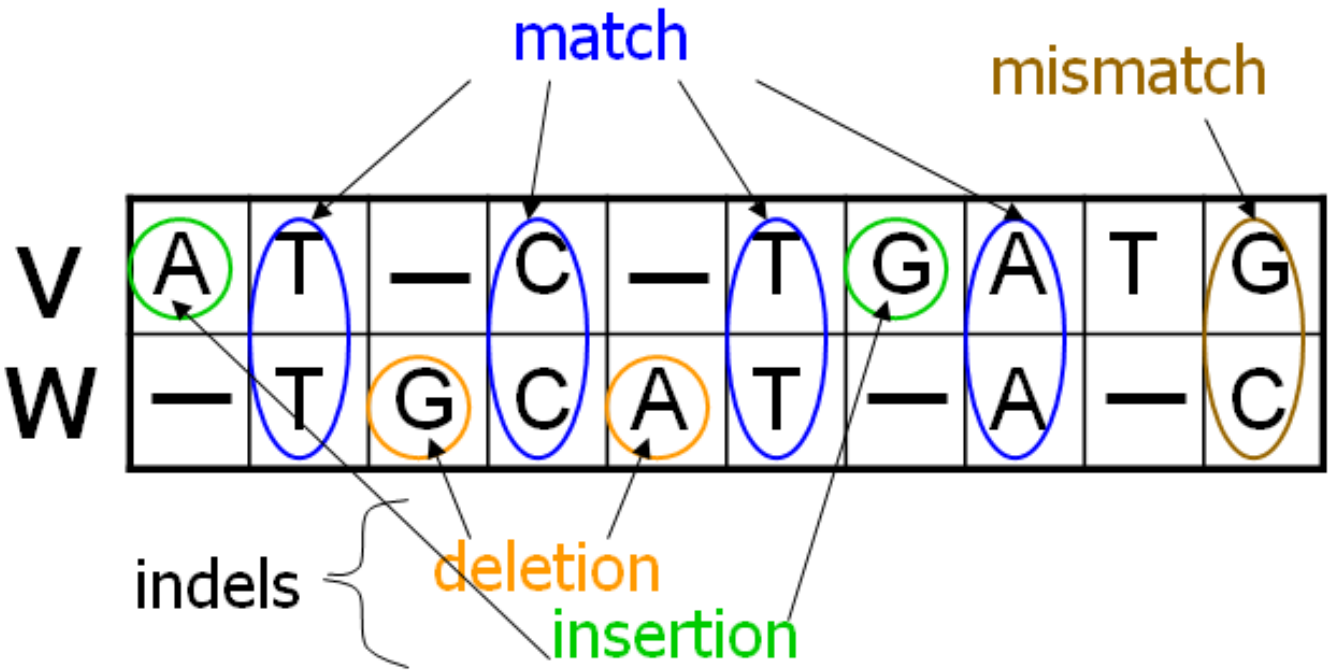
Motivación

- Fibrosis Cística: Cómo trabaja
  - La proteína CFTR (1480 aminoacidos) regula el *chloride ion channel*
  - Regula la "licuación" de los fluidos secretados por la célula
  - Personas con CF no tienen un aminoácido en el CFTR
  - Mucosidad término siendo muy densa, afectando muchos órganos

Comparación de secuencias y alineamiento

- ¿Cómo se mide la similitud de secuencias de ADN o la distancia entre secuencias?
  - **Distancia Hamming**: muy usado en Computación, pero, ¿cómo se comporta en secuencias de ADN?
    - Def: número de posiciones que difieren
    - ¿Qué sucede con ATATATAT y TATATATA?
  - **Distancia de Edición**: Levenshtein introdujo el concepto de edición, como el número de operaciones de edición necesarias para ir de v a w.
    - Def: operaciones indel (insert, deletions) y substituciones.
    - Calcular distancia de edición entre TGCATAT y ATCCGAT.
- **Edit distance** permite comparar strings de diferente tamaño.

Comparación de secuencias y alineamiento



Longest Common Subsequence (LCS)

- Es la forma más simple de alineamiento, solo permite *indels*, pero no reemplazos (*mismatches*)
- En este problema se asigna un 1 para los aciertos y un 0 para los *indels*
- Subsecuencia de v es una secuencia ordenada de caracteres (no necesariamente consecutivas) de v
  - Por ejemplo si $v = ATTGCTA \Rightarrow$
    - $AGCA$ y $ATTA$ son subsecuencias de v
    - pero $TGTT$ y TCG no lo son
- Formalmente, dado $v = v_1 ... v_n$ y $w = w_1 ... w_m$, una subsecuencia común a v y w se define por
  - una secuencia de posiciones en v, $1 \leq i_1 < i_2 < ... < i_k \leq n$ y
  - una secuencia de posiciones en w, $1 \leq j_1 < j_2 < ... < j_k \leq m$
  - tal que los símbolos en las posiciones correspondientes en v y w son iguales

$$v_{it} = w_{it}, 1 \leq t \leq k$$

  - Por ejemplo, $TCTA$ es una subsecuencia de $A\mathbf{TCT}G\mathbf{A}T$ y $\mathbf{T}G\mathbf{C}A\mathbf{TA}$

Longest Common Subsequence

- Pueden existir muchas subsecuencias comunes entre dos strings. ¿Cómo encontrar la subsecuencia más larga?
- Sea $s(v, w)$ el largo de LCS de v y w, entonces la distancia de edición (solo indels) será

$$d(v, w) = n + m - 2s(v, w)$$

- corresponde al número mínimo de *indels* necesarios para transformar v en w.
- ¿Cómo calcular el largo del LCS?
  - Sea v_i el prefijo de v de largo i
  - Sea w_j el prefijo de w de largo j
  - El largo de $LCS(v_i, w_j)$ es:

$$s_{i,j} = max \begin{cases} s_{i-1,j} \\ s_{i,j-1} \\ s_{i-1,j-1} + 1 \quad \text{si } v_i = w_j \end{cases}$$

Longest Common Subsequence

- Algoritmo

$LCS(v, w)$

1 **for** $i = 0$ **to** n

2 $\quad s_{i,0} \leftarrow 0$

3 **for** $j = 0$ **to** m

4 $\quad s_{0,j} \leftarrow 0$

5 **for** $i = 0$ **to** n

6 $\quad$ **for** $j = 0$ **to** m

7 $$s_{i,j} = max \begin{cases} s_{i-1,j} \\ s_{i,j-1} \\ s_{i-1,j-1} + 1 \quad \text{si } v_i = w_j \end{cases}$$

8 $$b_{i,j} = \begin{cases} \uparrow & \text{si } s_{i,j} = s_{i-1,j} \\ \leftarrow & \text{si } s_{i,j} = s_{i,j-1} \\ \nwarrow & \text{si } s_{i,j} = s_{i-1,j-1} + 1 \end{cases}$$

9 **return** $(s_{n,m}, b)$

Longest Common Subsequence

- Algoritmo para reconstruir la secuencia

```
    PrintLCS(b,v,i,j)
1       if i = 0 or j = 0
2           return
3       if b_{i,j} =↖
4           PrintLCS(b,v,i-1,j-1)
5           print v_i
6       else
7           if b_{i,j} =↑
8               PrintLCS(b,v,i-1,j)
9           else
10              PrintLCS(b,v,i,j-1)
```

Longest Common Subsequence

- Encontrar la LCS de las siguientes secuencias: (v=ATCTGAT,w=TGCATA) y (v=ATGTTAT, w=ATCGTAC)
- Calcular la similaridad y la distancia

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

- Calculando la similaridad.

- Calculando la similaridad.

	-	T	G	C	A	T	A
-	0	0	0	0	0	0	0
A	**0**	↑ 0	↑ 0	↑ 0	← 1	← 1	↖ 1
T	0	↖ **1**	← **1**	← 1	↑ 1	↖ 2	← 2
C	0	↑ 1	↑ 1	↖ **2**	← **2**	↑ 2	↑ 2
T	0	↖ 1	↑ 1	↑ 2	↑ 2	↖ **3**	← 3
G	0	↑ 1	↖ 2	↑ 2	↑ 2	↑ **3**	↑ 3
A	0	↑ 1	↑ 2	↑ 2	↖ 3	↑ 3	↖ **4**
T	0	↖ 1	↑ 2	↑ 2	↑ 3	↖ 4	↑ **4**

- Calculando la distancia de edición.

- Calculando la distancia de edición.

	-	T	G	C	A	T	A
-	0	1	2	3	4	5	6
A	**1**	↑ 2	↑ 3	↑ 4	↖ 3	← 4	↖ 5
T	2	↖ **1**	← **2**	← 3	↑ 4	↖ 3	← 4
C	3	↑ 2	↑ 3	↖ **2**	← **3**	↑ 4	↑ 5
T	4	↖ 3	↑ 4	↑ 3	↑ 4	↖ **3**	← 4
G	5	↑ 4	↖ 3	↑ 4	↑ 5	↑ **4**	↑ 5
A	6	↑ 5	↑ 4	↑ 5	↖ 4	↑ 5	↖ **4**
T	7	↖ 6	↑ 5	↑ 6	↑ 5	↖ 4	↑ **5**

Alineamiento de Secuencias

- El problema de LCS es la forma más simple de alineamiento, solo permite *indels*, pero no reemplazos (*mismatches*)
- En este problema se asigna un 1 para los aciertos y un 0 para los *indels*
- Se puede agregar una penalización a los indels y reemplazos
- El esquema más simple puede ser:
  - $+1$: acierto
  - $-\mu$: penalidad de reemplazo
  - $-\sigma$: penalidad de indel
- Se puede usar una matriz de puntuación para medir las diferentes operaciones. El puntaje final representa el puntaje del alineamiento
- El score resultante será entonces:

$$\#aciertos - \mu(\#reemplazos) - \sigma(\#indels)$$

Problema de Alineamiento Global

- Alineamiento Global vs Alineamiento local
  - El problema de Alineamiento Global intenta encontrar el camino más largo en la matriz de edición desde la posición (0,0) a la posición (n,m)
  - El problema de Alineamiento Local, intenta encontrar el caminos más largo entre 2 vértices de la matriz de edición
- Debido a los factores negativos de puntuación, es posible que un alineamiento local tenga un score más grande que uno global

Alineamiento Global vs Local

- Alineamiento Global

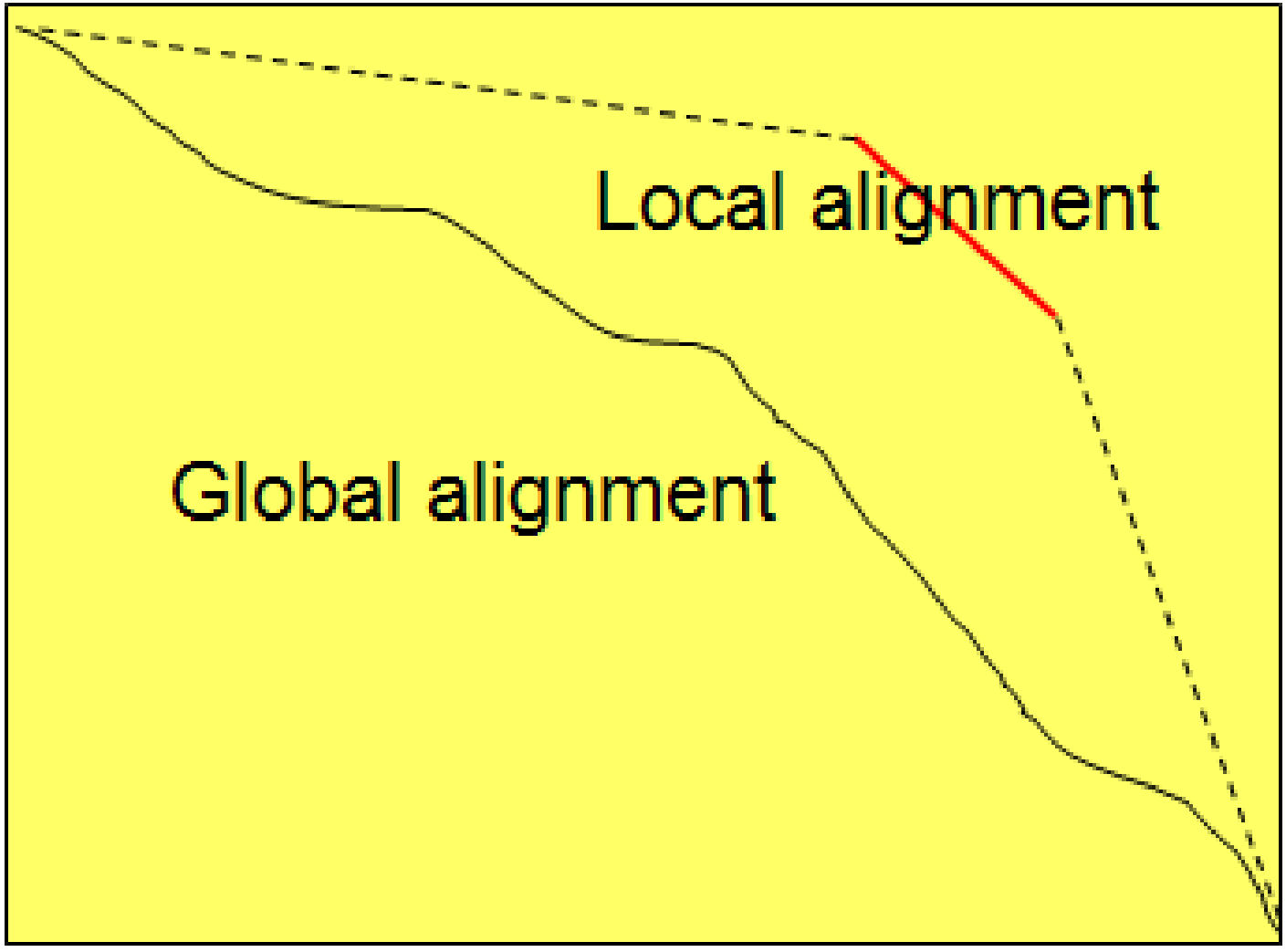
```
--T--CC-C-AGT--TATGT-CAGGGGACACG-A-GCATGCAGA-GAC
  |  || |  ||  | | | |||    || | | |  | ||||   |
AATTGCCGCC-GTCGT-T-TTCAG----CA-GTTATG-T-CAGAT--C
```

- Alineamiento Local

```
                    tccCAGTTATGTCAGgggacacgagcatgcagagac
                       ||||||||||||
aattgccgccgtcgttttcagCAGTTATGTCAGatc
```

Alineamiento Global vs Local

- Ejemplo



Problema de Alineamiento Global

- Encontrar el mejor alineamiento entre 2 strings considerando algún esquema de puntuación
  - **INPUT**: Strings v y w y un esquema de puntuación
  - **OUTPUT**: Alineamiento con la puntuación máxima entre todos los alineamientos posibles

$$
\begin{aligned}
\uparrow \rightarrow &= -\sigma \\
\nwarrow &= 1 \text{ (acierto)} \\
\nwarrow &= -\mu \text{ (reemplazo)}
\end{aligned}
$$

$$
s_{i,j} = max \begin{cases} s_{i-1,j} - \sigma \\ s_{i,j-1} - \sigma \\ s_{i-1,j-1} - \mu & \text{si } v_i \neq w_j \\ s_{i-1,j-1} + 1 & \text{si } v_i = w_j \end{cases}
$$

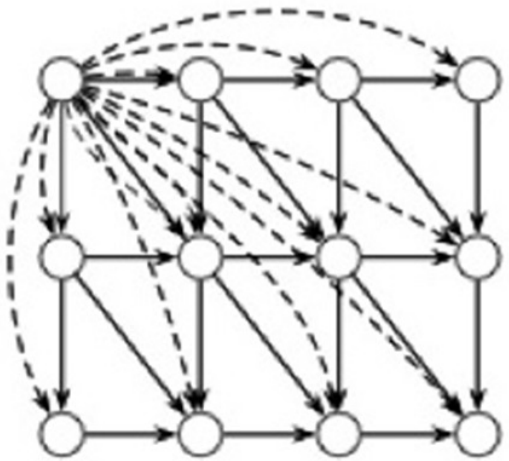
- LCS corresponde al problema de Alineamiento Global con parámetros $\sigma = 0, \mu = 0$ (o $\mu = \infty$)

Alineamiento Local

- ¿Por qué es útil?
  - Dos genes de especies diferentes pueden ser similares solo en pequeñas regiones conservadas y distintos en el resto
- Definición del problema:
  - Objetivo: Encontrar el mejor alineamiento global entre dos strings
  - Entrada: Strings v, w y matriz de penalizaciones δ
  - Salida: Alineamiento de substrings de v y w con la mejor puntuación entre **TODOS LOS ALINEAMIENTOS POSIBLES** de **TODOS LOS POSIBLES SUBSTRINGS**
- Se transforma en un problema computacionalmente costoso a medida que crece el tamaño de los strings.
- Alternativa de solución: Algoritmo de Smith-Waterman (1981)

Alineamiento Local

- Algoritmo de Smith-Waterman
  - Variación del algoritmo de Needleman-Wunsch (alineamiento global)
  - Utiliza programación dinámica $\Rightarrow$ encuentra el alineamiento local óptimo con respecto a la matriz de penalizaciones dada
  - En pocas palabras, el algoritmo incluye aristas de costo 0, desde el nodo inicial a cada nodo del grafo



Alineamiento Local

- Algoritmo de Smith-Waterman
  - Dada un matriz de penalizaciones δ

$$s_{i,0} = 0 \quad 0 \leq i \leq n$$
$$s_{0,j} = 0 \quad 0 \leq j \leq m$$
$$s_{i,j} = max \begin{cases} 0 & \text{(CAMBIO INTRODUCIDO)} \\ s_{i-1,j} + \delta(v_i, -) & \textit{Deletion} \\ s_{i,j-1} + \delta(-, w_j) & \textit{Insert} \\ s_{i-1,j-1} + \delta(v_i, w_j) & \textit{Acierto/Reemplazo} \end{cases}$$

Alineamiento Local

- El alineamiento local solo reporta la secuencia más larga encontrada
- En ocasiones diversos alineamientos locales pueden tener diversos significados biológicos
- Se han desarrollado métodos para encontrar los k mejores alineamientos no traslapados
- Métodos particularmente importantes para comparaciones de proteinas de múltiples dominios
  - presencia de bloques similares
  - bloques han sido movidos de un lugar a otro

Alinamiento con Penalidades por Gaps

- Mutaciones son causadas usualmente por errores en el proceso de replicación
- Se producen inserciones o borrados de substrings enteros en vez de nucleotidos individuales
- **Gap**: es definido como secuencia de espacios en una de las filas del alineamiento
- ¿Qué sucede con la penalización de un gap de largo x?
- el usar $x \times \sigma$ no es correcto porque perjudica a estos gaps
- se redefine la función de ponderación de un gap como $-(\rho + \sigma x)$
  - con ρ ponderación por la presencia de un gap y σ es la penalización por cada elemento del gap
  - $\rho >> \sigma$

Alineamiento de Múltiples Secuencias

- Alineamiento de pares puede no detectar relaciones bioógicas débiles entre secuencias
- Alinear varias secuencias a la vez ha mostrado la capacidad de indentificar similaridades que pasan desapercibidas
- Problema
  - Sea $v_1, ..., v_k$ secuencias de largo $n_1, ..., n_k$ sobre un alfabeto $A + \{-\}$ extendido
  - El alineamiento múltiple es una matriz de $k \times n$ con $n \geq max(n_1, .., n_k)$
  - Cada fila es una de las palabras v_i con los caracteres en orden y $n - n_i$ espacios
  - Cada columna contiene al menos un símbolo de A (no hay columnas vacías)
- Una solución es usar el mismo algoritmo anterior pero para k secuencias

Alineamiento de Múltiples Secuencias

- Queremos encontrar el mejor alineamiento entre las secuencias u, v y w
- Se puede aplicar la misma lógica usada que para dos secuencias
- Se crea una matriz de programación dinámica de tres dimensiones

$$s_{i,j,k} = max \begin{cases} s_{i-1,j,k} & +\delta(v_i, -, -) \\ s_{i,j-1,k} & +\delta(-, w_j, -) \\ s_{i,j,k-1} & +\delta(-, -, u_k) \\ s_{i-1,j-1,k} & +\delta(v_i, w_j, -) \\ s_{i-1,j,k-1} & +\delta(v_i, -, u_k) \\ s_{i,j-1,k-1} & +\delta(-, w_j, u_k) \\ s_{i-1,j-1,k-1} & +\delta(v_i, w_j, u_k) \end{cases}$$

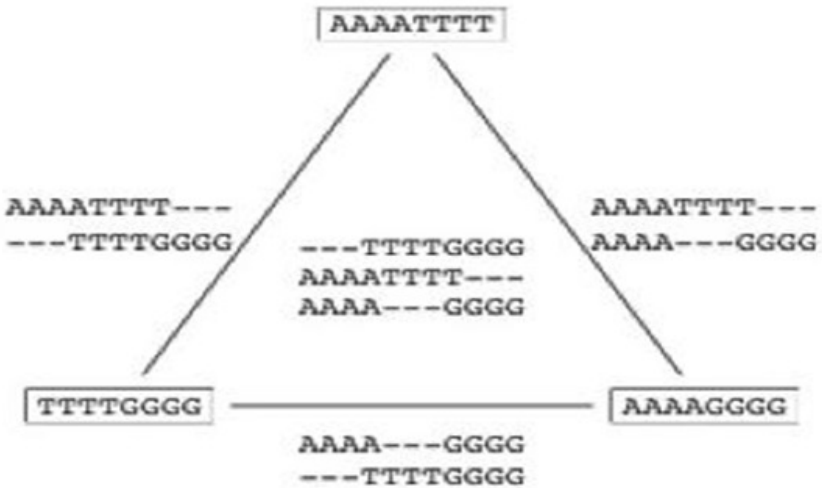
- En el caso de k secuencias, la complejidad algoritmica es $O(2n)^k$

Alineamiento de Múltiples Secuencias

- ¿Es factible hacer alineamientos de pares y luego combinarlos?

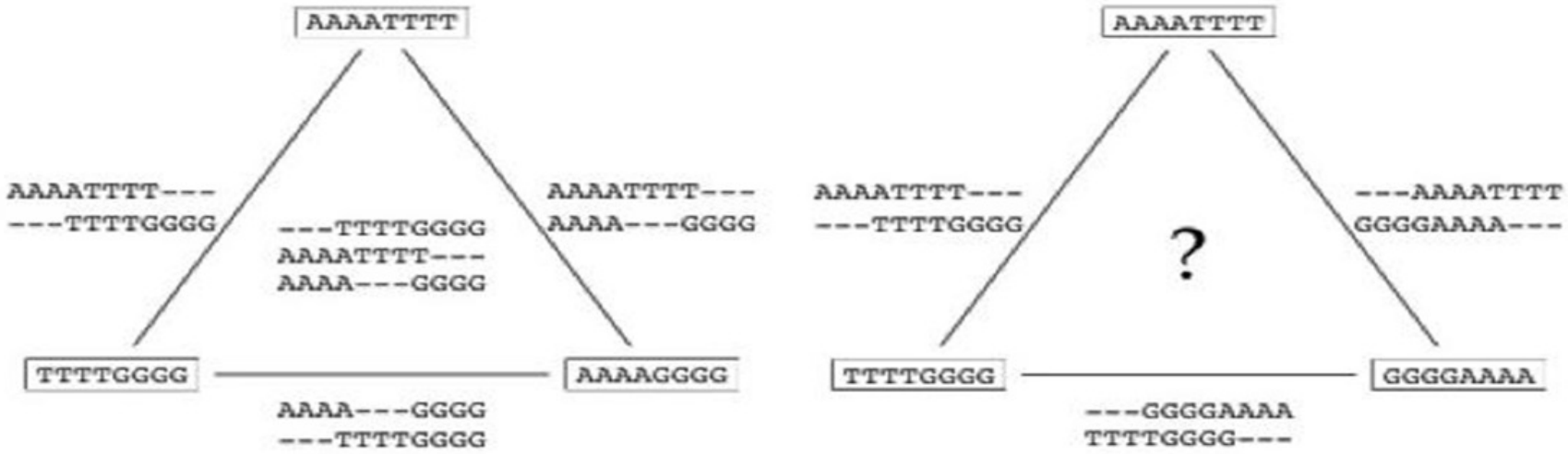
Alineamiento de Múltiples Secuencias

- ¿Es factible hacer alineamientos de pares y luego combinarlos?



Alineamiento de Múltiples Secuencias

- ¿Es factible hacer alineamientos de pares y luego combinarlos?



- Otro enfoque es usar "multiple k-way alignment".
- Mezclar los dos mejores alineamientos de pares y crear una única secuencia.
- Realizar múltiple alineamiento entre $k-1$ secuencias. Enfoque seguido por CLUSTAL.

Alineamiento de Múltiples Secuencias

- ¿Cómo asignar un valor a un alineamiento múltiple?

Alineamiento de Múltiples Secuencias

- ¿Cómo asignar un valor a un alineamiento múltiple?
  - **Número de matches (multiple longest common subsequence)**: uno para cada columna con el mismo símbolo, cero en caso contrario
  - **Basado en entropía**: p_X: probabilidad de X en cada columna. La entropia del alineamiento se calcula como la suma de las entropias de cada columna. $(-\sum_{x\in(A,C,T,G)} p_x log\, p_x)$
  - **Suma de pares (SP-Score)**: cada alineamiento múltiple fuerza un alineamiento de pares. $SP = \sum_{i,j=1}^{k} score(v_i, v_j)$

Alineamiento de Múltiples Secuencias

- Ejemplo de puntuación

	A	A	A
	A	A	A
	A	A	T
	A	T	C
LCS			
Entropy			
SP-pairs			

Matrices de Penalización

- Para generalizar la puntuación considere una matriz δ de (4+1)x(4+1) para nucleotidos.
- En el caso de aminoácidos la matriz δ sería de (20+1)x(20+1)
- Se agrega 1 para contabilizar los *indels*
- ¿Cómo medir la similaridad entre dos secuencias?
  - Basado en porcentaje de similaridad de las secuencias
  - Basado en conservación

Matrices de Penalización

- Son creadas en base a la evidencia biológica
- Alineamientos pueden ser vistos como dos secuencias que son distintas debido a sus mutaciones
- Algunas de estas mutaciones tienen un efecto despreciable en la función de la proteína, por lo que las penalidades serán distintas entre ellas

Matrices de Penalización

- Ejemplo

	A	R	N	K
A	5	-2	-1	-1
R	-	7	-1	3
N	-	-	7	0
K	-	-	-	6

AKRANR

KAAANK

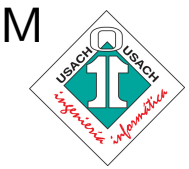
-1 + (-1) + (-2) + 5 + 7 + 3 = 11

Conservación

- Algunos reemplazos en los aminoacidos tienden a preservar las propiedades físico-químicas del residuo original
  - Aspartate $\rightarrow$ Glutamate
  - Alanine $\rightarrow$ Valine
  - Leucine $\rightarrow$ Isoleucine
- Existen varias matrices de sustitución de aminoacidos
  - PAM: Point Accepted Mutation (Dayhoff et al)
  - BLOSUM: Blocks Substitution Matrix
- Matrices de sustitución de ADN
  - ADN es menos conservativo que las secuencias de proteínas
  - Por lo tanto es menos efectivo comparar regiones de código a nivel de nucleotidos

PAM (Point accepted mutations)

- Depende de la similitud de las secuencias y distancia evolutiva de las especies
- Primero se analizan proteinas muy parecidas (una mutación cada 100 aminoacidos en promedio)
- PAM 1: número de veces que una proteína promedio muta el 1% de sus aminoacidos
- Para calcular PAM se define:
  - $f(i,j)$ = número de veces que el aminoacido i es alineado con el aminoacido j, dividido por el largo del alineamiento.
  - $f(i)$ = frecuencia del aminoacido i en todas las secuencias
  - $g(i,j) = \frac{f(i,j)}{f(i)}$
  - probabilidad que i mute en j en una unidad PAM.
  - $\text{PAM}(i,j) = log\frac{f(i,j)}{f(i)f(j)} = log\frac{g(i,j)}{f(j)}$
  - PAM_n indica la probabilidad de i mutar en j durante n unidades PAM



PAM (Point accepted mutations)

- Porcentaje promedio de aminoacidos que cambian en secuencias evolutivamente relacionadas
- $\text{PAM}_x = \text{PAM}_1^x$: aplicar la matriz PAM x veces (multiplicarla por si misma)
- Ejemplo $\text{PAM}_{250} = \text{PAM}_1^{250}$

	Ala	Arg	Asn	Asp	Cys	Gln	Glu	Gly	His	Ile	Leu	Lys	...
	A	R	N	D	C	Q	E	G	H	I	L	K	...
Ala A	13	6	9	9	5	8	9	12	6	8	6	7	...
Arg R	3	17	4	3	2	5	3	2	6	3	2	9	
Asn N	4	4	6	7	2	5	6	4	6	3	2	5	
Asp D	5	4	8	11	1	7	10	5	6	3	2	5	
Cys C	2	1	1	1	52	1	1	2	2	2	1	1	
Gln Q	3	5	5	6	1	10	7	3	7	2	3	5	
...													
Trp W	0	2	0	0	0	0	0	0	1	0	1	0	
Tyr Y	1	1	2	1	3	1	1	1	3	2	2	1	
Val V	7	4	4	4	4	4	4	4	5	4	15	10	

BLOSUM

- Valores derivados de la observación de frecuencias de sustitución en bloques de alineamientos locales en proteínas relacionadas
- El nombre de la matriz indica la distancia evolutiva
  - Ejemplo: BLOSUM50 fue creada usando secuencias que comparten no más del 50% de similaridad

	A	R	N	D	C	Q	E	G	H	I	L	K	M	F	P	S	T	W	Y	V	B	Z	X	*
A	5	-2	-1	-2	-1	-1	-1	0	-2	-1	-2	-1	-1	-3	-1	1	0	-3	-2	0	-2	-1	-1	-5
R	-2	7	-1	-2	-4	1	0	-3	0	-4	-3	3	-2	-3	-3	-1	-1	-3	-1	-3	-1	0	-1	-5
N	-1	-1	7	2	-2	0	0	0	1	-3	-4	0	-2	-4	-2	1	0	-4	-2	-3	4	0	-1	-5
D	-2	-2	2	8	-4	0	2	-1	-1	-4	-4	-1	-4	-5	-1	0	-1	-5	-3	-4	5	1	-1	-5
C	-1	-4	-2	-4	13	-3	-3	-3	-3	-2	-2	-3	-2	-2	-4	-1	-1	-5	-3	-1	-3	-3	-2	-5
Q	-1	1	0	0	-3	7	2	-2	1	-3	-2	2	0	-4	-1	0	-1	-1	-1	-3	0	4	-1	-5
E	-1	0	0	2	-3	2	6	-3	0	-4	-3	1	-2	-3	-1	-1	-1	-3	-2	-3	1	5	-1	-5
G	0	-3	0	-1	-3	-2	-3	8	-2	-4	-4	-2	-3	-4	-2	0	-2	-3	-3	-4	-1	-2	-2	-5
H	-2	0	1	-1	-3	1	0	-2	10	-4	-3	0	-1	-1	-2	-1	-2	-3	2	-4	0	0	-1	-5
I	-1	-4	-3	-4	-2	-3	-4	-4	-4	5	2	-3	2	0	-3	-3	-1	-3	-1	4	-4	-3	-1	-5
L	-2	-3	-4	-4	-2	-2	-3	-4	-3	2	5	-3	3	1	-4	-3	-1	-2	-1	1	-4	-3	-1	-5
K	-1	3	0	-1	-3	2	1	-2	0	-3	-3	6	-2	-4	-1	0	-1	-3	-2	-3	0	1	-1	-5
M	-1	-2	-2	-4	-2	0	-2	-3	-1	2	3	-2	7	0	-3	-2	-1	-1	0	1	-3	-1	-1	-5
F	-3	-3	-4	-5	-2	-4	-3	-4	-1	0	1	-4	0	8	-4	-3	-2	1	4	-1	-4	-4	-2	-5
P	-1	-3	-2	-1	-4	-1	-1	-2	-2	-3	-4	-1	-3	-4	10	-1	-1	-4	-3	-3	-2	-1	-2	-5
S	1	-1	1	0	-1	0	-1	0	-1	-3	-3	0	-2	-3	-1	5	2	-4	-2	-2	0	0	-1	-5
T	0	-1	0	-1	-1	-1	-1	-2	-2	-1	-1	-1	-1	-2	-1	2	5	-3	-2	0	0	-1	0	-5
W	-3	-3	-4	-5	-5	-1	-3	-3	-3	-3	-2	-3	-1	1	-4	-4	-3	15	2	-3	-5	-2	-3	-5
Y	-2	-1	-2	-3	-3	-1	-2	-3	2	-1	-1	-2	0	4	-3	-2	-2	2	8	-1	-3	-2	-1	-5
V	0	-3	-3	-4	-1	-3	-3	-4	-4	4	1	-3	1	-1	-3	-2	0	-3	-1	5	-4	-3	-1	-5
B	-2	-1	4	5	-3	0	1	-1	0	-4	-4	0	-3	-4	-2	0	0	-5	-3	-4	5	2	-1	-5
Z	-1	0	0	1	-3	4	5	-2	0	-3	-3	1	-1	-4	-1	0	-1	-2	-2	-3	2	5	-1	-5
X	-1	-1	-1	-1	-2	-1	-1	-2	-1	-1	-1	-1	-1	-2	-2	-1	0	-3	-1	-1	-1	-1	-1	-5
*	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	1

Actividad

- Dada la complejidad algoritmica y el tamaño de los datos, los algoritmos visto no son practicables. De esto nace BLAST (Basic Local Alignment Search Tool)
  - ¿Cómo trabaja BLAST?
  - ¿Cuál es la diferencia entre megablast, blastn, y discontinuos blast?
  - ¿Qué otros tipos de herramientas BLAST existen?
- Buscar la secuencia que se encuentra en la página del curso
- Usando las herramientas de NCBI y EBI responda las siguientes preguntas:
  - ¿A qué gen representa?
  - ¿Qué información puede encontrar sobre la secuencia? (organismo, año, etc)
  - ¿Es responsable por alguna enfermedad? (TIP: use la página de OMIM (*www.omim.org*)
  - ¿Hay alguna diferencia en la información proporcionada por NCBI y EBI?