Programación dinámica: algoritmo de Needleman-Wunsch y alineamientos pareados globales

Saul Needleman and Christian Wunsch (1970). *A general method applicable to the search for similarities in the amino acid sequence of two proteins*, J Mol Biol. **48**(3):443-53.

Este algoritmo es un ejemplo de PD y **garantiza encontrar el alineamiento global de puntuación máxima**

La PD constituye una técnica muy general de programación. Se suele aplicar cuando existe un espacio de búsqueda muy grande y éste puede ser estructurado en una serie o sucesión de estados tales que:

1. el estado inicial contiene soluciones triviales de subproblemas
2. cada solución parcial de estados posteriores puede ser calculada por iteración sobre un número fijo de soluciones parciales de los estados anteriores
3. el estado final contiene la solución final

Un algoritmo de PD consta de 3 fases:

1. fase de inicialización y definición recurrente del score óptimo
2. relleno de la matriz de PD para guardar los scores de subproblemas resueltos en cada iter. Se comienza por resolver el subproblema más pequeño
3. un rastreo reverso de la matriz para recuperar la estructura de la solución óptima

Programación dinámica y la generación de alineamientos pareados (globales y locales) - algoritmo de DP para alineamientos globales

• Como ejemplo vamos a alinear dos palabras: COELACANTH y PELICAN usando el siguiente esquema de ponderación: match = 1; mismatch = -1; gap = -1

Existen dos alineamientos con el mismo score máximo:

```
COELACANTH        COELACANTH
P-ELICAN--   y    -PELICAN--
```

por tratarse de aln. globales, cada letra está alineada con otra o con un gap. Este no es el caso en aln. locales.

El alineamiento acontece en un arreglo bidimensional en el que cada celda corresponde al apareamiento de un resíduo de cada secuencia

El alineamiento comienza arriba izda y sigue una trayectoria horizontal o vertical cuando hay un gap que introducir, y en la diagonal cuando tenemos apareamientos. Los gaps nunca se aparean entre ellos

Nótese que tenemos una fila y col. vacías adicionales

Programación dinámica: algoritmo de Needleman-Wunsch y alineamientos pareados globales

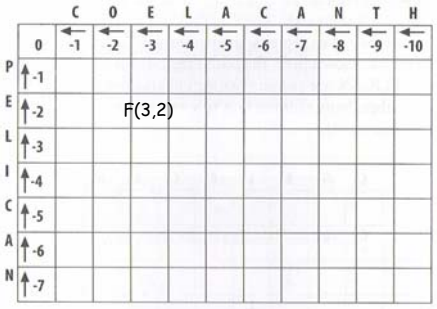
En realidad no se gurdan los caracteres en las celdas. Estas contienen dos valores: una puntuación (score) y un apuntador. El score se calcula a partir del esquema de puntuación o más generalmente, de una matriz de puntuaciones. El apuntador es un indicador de dirección (flecha) que apunta en una de tres direcciones: arriba, izquierda o en diagonal izda. hacia arriba.

I. Fase de inicialización

• se comienza asignando valores a la primera fila y columna. La siguiente fase del algoritmo depende de estas asignaciones.

• La puntuación de cada celda corresponde al "gap score" x distancia al origen

• Las flechas apuntan todas al origen, lo que asegura que los alineamientos puedan seguirse hasta el origen al final del algoritmo. Esto es un requisito para conseguir un aln. global



F(i, 0) = i x gap penalty; i = pos columna
F(0, j) = j x gap penalty j = pos fila

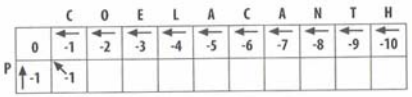
Programación dinámica: algoritmo de Needleman-Wunsch y alineamientos pareados globales

II. Fase de relleno o inducción.

- Se rellena toda la tabla con "scores" y apuntadores, requiréndose los valores de las celdas vecinas diagonal, vertical y horizontal. Por ello sólo se puede comenzar en la celda (1,1)
- Se calculan tres scores: uno de match, uno de gap horizontal y otro de gap vertical:

1. El **match score** = score de la diagonal + puntuación de apareamiento (+1 ó -1)
2. El **gap score horizontal** = score de celda izda + gap score
3. El **gap score vertical** = score de celda superior + gap score
4. Se asigna a la nueva celda el valor más alto de los tres y una flecha en dirección de la celda vecina con mayor score

$$F(i,j) = \begin{cases} F(i-1, j) + \text{gap-penalty} \\ F(i-1, j-1) + s(i,j) \\ F(i, j-1) + \text{gap-penalty} \end{cases}$$



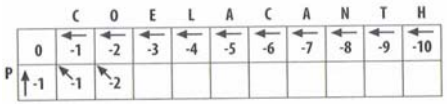
1. match score = 0 + (-1) = -1 ➔ es el score más alto y por tanto va a la celda
2. gap score horizontal = -1 + (-1) = -2
3. gap score vertical = -1 + (-1) = -2
4. la flecha apunta al 0 por ser el score vecino más alto

Programación dinámica: algoritmo de Needleman-Wunsch y alineamientos pareados globales

II. Fase de relleno o inducción.

- Segundo ciclo. Se continúa llenando la segunda fila o columna siguiendo las mismas reglas

$$F(i,j) = \begin{cases} F(i-1, j) + \text{gap-penalty} \\ F(i-1, j-1) + s(i,j) \\ F(i, j-1) + \text{gap-penalty} \end{cases}$$



El mejor score del alineamiento hecho hasta ahora tiene vale -2 y corresponde a:

```
CO      CO
-P  ó   P-
```

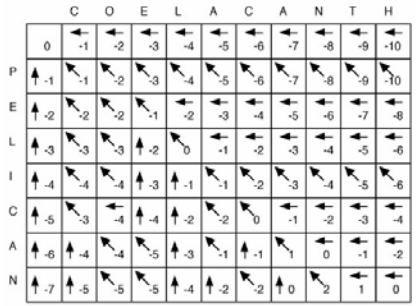
1. match score = -1 + (-1) = -2 ➔ es el score más alto y por tanto va a la celda
2. gap score horizontal = -1 + (-1) = -2
3. gap score vertical = -2 + (-1) = -3
4. la flecha puede apuntar al -1 de la diagonal u horizontal. Se toma una decisión arbitraria pero consistente si se vuelve a dar el caso (p. ej. aceptar siempre diagonal).

Programación dinámica: algoritmo de Needleman-Wunsch y alineamientos pareados globales

II. Fase de relleno o inducción.

- Segundo ciclo. Se continúa llenando la segunda fila (o columna) siguiendo las mismas reglas y una vez llena, se continúa con la tercera fila (o columna) hasta terminar de llenar la tabla siguiendo la expresión:

$$F(i,j) = \max\{F(i-1, j-1) + s(i,j),\ F(i-1, j) + \text{gap-penalty},\ F(i, j-1) + \text{gap-penalty}\}$$

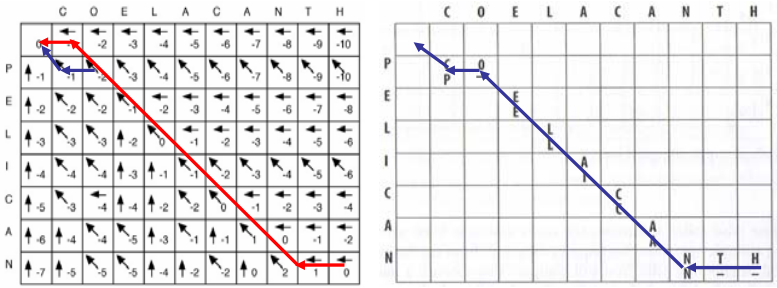


Programación dinámica: algoritmo de Needleman-Wunsch y alineamientos pareados globales

III. Fase de rastreo regresivo o hacia el origen

Para recuperar el alineamiento tenemos que regresarnos de la celda ubicada en el vértice de abajo a la dcha. y seguir el camino indicado por el puntero hasta el inicio

Dado que seguimos el camino del alineamiento óptimo del final hacia el principio, tenemos que revertir la secuencia al final del algoritmo para tenerla en la orientación correcta



Existen dos alineamientos globales con el mismo score máximo = 0

```
COELACANTH      COELACANTH
-PELICAN--  y   P-ELICAN--
```

Por escojer la opción de seguir diagonal sólo obtenemos **uno**

Programación dinámica: algoritmo de Smith-Waterman y alineamientos pareados locales

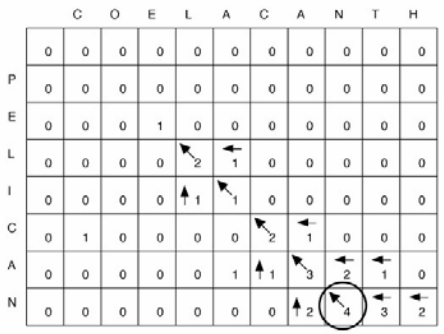
Smith TF, Waterman MS (1981) J. Mol. Biol 147(1);195-7

- Se trata de una modificación simple del algoritmo de Needleman-Wunsch. Sólo hay **tres cambios:**

1. **La 1a. fila y columna es inicializada con ceros**, en vez de gap penalties incrementales
2. **El score máximo no es nunca < 0 y sólo se guardan apuntadores en las celdas si su score es > 0**
3. **El rastreo reverso comienza desde la celda con el score más alto de la tabla** (y no de la última celda de la misma) **y termina en una celda con score 0** (y no en la primera)

- Estas modificaciones tienen un profundo efecto sobre el comportamiento del algoritmo, y **como resultado obtenemos el alineamiento local con mayor puntuación** de todos los posibles en la matriz.

Programación dinámica: algoritmo de Smith-Waterman y alineamientos pareados locales



El alineamiento local con el máximo score = 4 es:

ELACAN
ELICAN

Programación dinámica: Notas prácticas sobre el uso de los algoritmos de Smith-Waterman y Needleman-Wunsh.

Alineamientos globales vs. locales

- Aunque muy similares desde el punto de vista mecanístico, ambos tienen propiedades y aplicaciones muy diferentes. Por ejemplo, si queremos alinear dos genes eucarióticos muy divergentes esperaríamos que la estructura y secuencia de exones esté relativamente conservada, si bien los intrones habrán sufrido muchos eventos de indel.
- Los exones tal vez sólo representen el 1-5% de la secuencia de estos genes. Por ello si queremos usar una estrategia de alineamiento global el resultado seguramente será desastroso desde un punto de vista biológico. Muy posiblemente las regiones exónicas homólogas no se alineen. Ello se debe a que su contribución a la puntuación (score) del alineamiento es mínimo dado su reducido tamaño relativo.
- En cambio un algoritmo de aln. local sí podrá identificar y alinear correctamente a las regiones exónicas homólogas. Pero usando implementaciones como las vistas en el ejemplo sólo recuperaremos aquel aln. local con la puntuación más alta.
- Estas limitaciones de los algoritmos clásicos de SW y NW han sido eliminadas en las múltiples variantes que existen de los mismos para distinto propósitos (BLAST, Clustal, etc).

Programación dinámica: Notas prácticas sobre el uso de los algoritmos de Smith-Waterman y Needleman-Wunsh.

- Como vimos en los ejemplos anteriores, durante la fase de llenado cada nueva celda rellenada representa el alineamiento con máxima puntuación entre el par de secuencias encontrados hasta dicho punto. Al calcular la siguiente celda, se emplean los valores previamente guardados. Por tanto la PD es una función de optimización cuya definición se extiende a medida que progresa el algoritmo.
- Los algoritmos de DP descritos tienen una complejidad $O(nm)$ tanto en tiempo como en memoria, donde n y m son la longitud de las secuencias a alinear. No se deben por tanto usar estos algoritmos para alinear secuencias largas como por ejemplo dos genomas. El no. de celdas requeridas es de $n \times m$ y cada celda toma unos 8 bytes de memoria. Por tanto, alinear dos secuencias de unas 100kb cada una demandaría unos 80 gibabytes (GB) de RAM.
- De ahí que se han desarrollado versiones de memoria linear (y no cuadrática) de estos algoritmos.

Programación dinámica: algoritmos de Smith-Waterman y Needleman-Wunsh ejercicios.

1°. Ir a la página del NCBI y descargar las secuencias de los citocromos C P00001 y P00090, y de las proteínas P13569 y P33593, en formato fasta

http://www.ncbi.nlm.nih.gov/

2°. Ir a la página del Instituto Pasteur en Paris y hacer un alineamiento global de los citocromos C P00001 y P00090 usando el programa NEEDLE del paquete EMBOSS

http://bioweb.pasteur.fr/seqanal/interfaces/needle.html

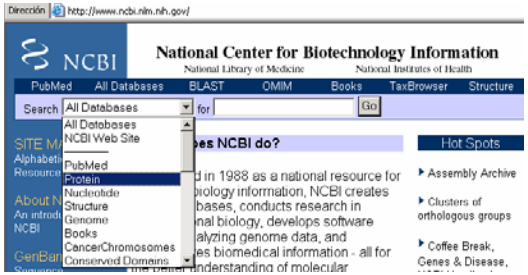
3°. Correr un alineamiento local con las proteínas P13569 y P33593 usando el programa WATER

http://bioweb.pasteur.fr/seqanal/interfaces/water.html

Programación dinámica: algoritmos de Smith-Waterman y Needleman-Wunsh ejercicios.

1°. Ir a la página del NCBI y descargar las secuencias de los citocromos *C* P00001 y P00090, y de las proteínas P13569 y P33593, en formato fasta

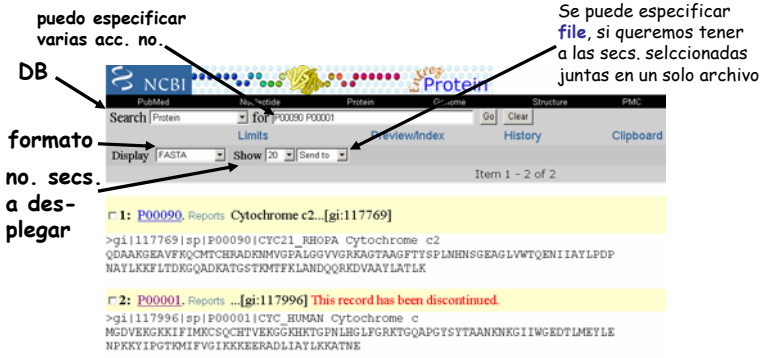
http://www.ncbi.nlm.nih.gov/



Programación dinámica: algoritmos de Smith-Waterman y Needleman-Wunsh ejercicios.

1°. Ir a la página del NCBI y descargar las secuencias de los citocromos *C* P00001 y P00090, y de las proteínas P13569 y P33593, en formato fasta

http://www.ncbi.nlm.nih.gov/



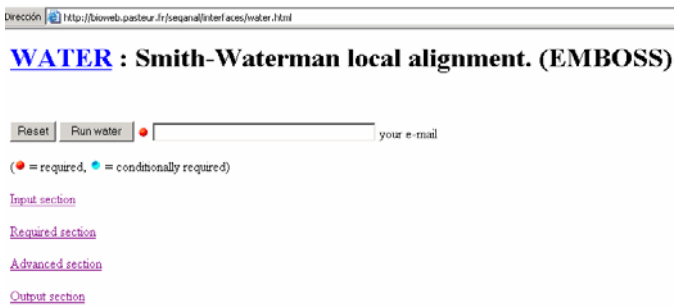
Programación dinámica: algoritmos de Smith-Waterman y Needleman-Wunsh ejercicios.

2°. Ir a la página del Instituto Pasteur en Paris y hacer un alineamiento global de los citocromos *C* P00001 y P00090 usando el programa NEEDLE del paquete EMBOSS

http://bioweb.pasteur.fr/seqanal/interfaces/needle.html

3°. Correr un alineamiento local con las proteínas P13569 y P33593 usando el programa WATER

http://bioweb.pasteur.fr/seqanal/interfaces/water.html



Programación dinámica: alogritmos de Smith-Waterman y Needleman-Wunsh ejercicios.

- Ejercicio: alinear **a mano** los oligonucleótidos **TTCATA** y **TGCTCGTA** usando el algoritmo de Needleman y Wunsch con el siguiente esquema de ponderación:
  match = +5; mismatch = -2; gap = -6

- Recuerda que en la práctica no se usan valores simples de penalización de gaps como los usados en nuestros ejemplos. Se usa un sistema de ponderación de gaps afines, con un valor de penalización de apertura de gap mayor que el de extensión: $w = g + hl$

- Además, para alinear (pares de) secuencias de proteínas se emplean matrices empíricas de costo de sustitución. Cómo se generan dichas matrices es lo que veremos en las siguientes páginas.