Biología Genómica y Evolución IV -
Inferencia Filogenética y Evolución Molecular

Semestre 2007-1

Pablo Vinuesa (vinuesa@ccg.unam.mx)

Progama de Ingeniería Genómica, CCG-UNAM, México
http://www.ccg.unam.mx/~vinuesa/

Todo el material del curso (presentaciones, lecturas, ejercicios, tutoriales, URLs ...) lo encontrarás en:
http://cursos.lcg.unam.mx/courses/BGE_IV_2007/

- **Tema 3: alineamientos pareados y búsqueda de homólogos en bases de datos**
  - evolución de secuencias y **clasificación de mutaciones**
  - **indeles** y **gaps**
  - **alineamientos globales** (Needleman-Wunsch) vs. **locales** (Smith-Waterman);
  - **programación dinámica**;
  - **dot plots**;
  - **matrices de costo de sustitución, penalización de gaps** y cuantificación de la similitud;
  - evaluación estadística de la similitud entre pares de secuencias;
  - escrutinio de bases de datos mediante **BLAST**; Búsquedas a nivel de **DNA vs. AA**;
  - la **familia BLAST** e interpretación de resultados de **búsqueda de secuencias homólogas**
  - prácticas: uso de **NCBI BLAST en línea**

Protocolo básico para un análisis filogenético de secuencias moleculares

Tema 3: alineamientos pareados, búsquedas de homólogos en bases de datos

Colección de secuencias homólogas

⇩ · BLAST y FASTA

Alineamiento múltiple de secuencias

⇩ · Clustal, T-Coffee, muscle...

Análisis evolutivo del alineamiento y selección del modelo de sustitución más ajustado

⇩ · tests de saturación, modeltest, ...

Estima filogenética

⇩ · NJ, ME, MP, ML, Bayes ...

Pruebas de confiabilidad de la topología inferida

⇩ · proporciones de bootstrap probabilidad posterior ...

Interpretación evolutiva y aplicación de las filogenias

Alineamientos pareados y búsqueda de homólogos en bases de datos

Los alineamientos pareados son la base de lo métodos de búqueda de secuencias homólogas en bases de datos

- Si dos proteínas o genes se parecen mucho a lo largo de toda su longitud asumimos que se trata de proteínas o genes homólogos, es decir, descendientes de un mismo ancestro común (cenancestro).
- Por ello una de las técnicas más utilizadas para detectar potenciales homólogos en bases de datos de secuencias se basa en la **cuantificación de la similitud entre pares de secuencias** y la determinación de la **significancia estadística** de dicho parecido. Estas magnitudes son las que reportan los estadísticos de **BLAST**.

```
>gi|71548896|ref|ZP_00669120.1|  Translation elongation factor G:Small GTP-binding protein domain
[Nitrosomonas eutropha C71]
 gi|71480607|gb|EAO18626.1|  Translation elongation factor G:Small GTP-binding protein domain
[Nitrosomonas eutropha C71]
Length=696

 Score =  828 bits (2140),  Expect = 0.0
 Identities = 434/697 (62%), Positives = 541/697 (77%), Gaps = 9/697 (1%)

Query  1    MTREFSLEKTRNIGIMAHIDAGKTTTTERVLYYTGRIHKIGETHEGASQMDWMAQEQERG  60
            M++   LE+ RNIGIMAHIDAGKTTT+ER+L+YTG  HK+GE H+GA+ MDWM QEQERG
Sbjct  1    MSKRNPLERYRNIGIMAHIDAGKTTTSERILFYTGVSHKLGEVHDGAATMDWMEQEQERG  60

Query  61   XXXXXXXXXXWN-------DHRINIIDTPGHVDFTVEVERSLRVLDGAVAVLDAQSGVE  113
            ITITSAATT  W        +HRIN+IDTPGHVDFT+EVERSLRVLDGA  V  +  GV+
Sbjct  61   ITITSAATTCFWKGMAGNYPEHRINVIDTPGHVDFTIEVERSLRVLDGACTVFCSVGGVQ  120
```

(... truncado)

Alineamiento de secuencias de DNA y proteína - introducción

- Dadas 2 o más secuencias, lo que generalmente deseamos es:

1. cuantificar su grado de similitud
2. determinar las correspondencias evolutivas (homología) residuo – residuo
3. describir e interpretar patrones de conservación y variación
4. inferir las relaciones evolutivas entre las secuencias

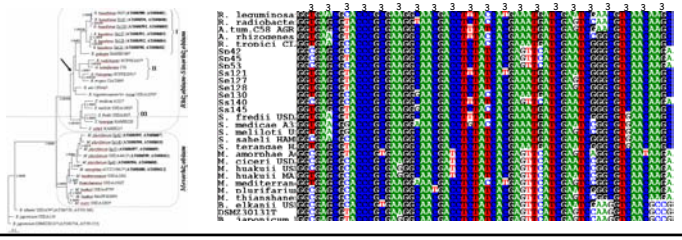
- Para definir índices cuantitativos de similitud entre secuencias necesitamos primero definir las **correspondencias evolutivas (homología) entre los residuos** de distintas secuencias, en forma de un **alineamiento**. Este representa una de las herramientas básicas de la bioinformática y biología evolutiva

- Para optimizar un alineamiento necesitamos acomodar las correspondencias entre resíduos idénticos, distintos, inserciones y deleciones. Esto se logra matemáticamente usando **factores de ponderación** ("weightings") para cada caso. Así un match tiene un peso, un mismatch otro y los indeles un tercer valor. Dos secuencias se comparan resíduo a resíduo, generándose un valor de puntuación (**score**) acorde a estas ponderaciones, que refleja el nivel de similitud entre ellas

Homología entre secuencias de DNA y proteína: conceptos y terminología básica

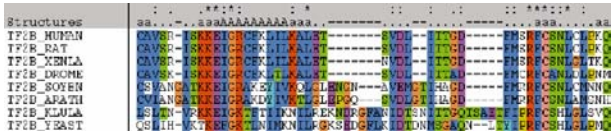
- A lo largo de la evolución las secuencias descendientes de otra ancestral van acumulando diversos tipos de mutaciones. Estas son **mutaciones puntuales** o **reorganizaciones genómicas**, que pueden involucrar **inserciones**, **deleciones**, **inversiones**, **translocaciones o duplicaciones**, mediados por distintos mecanismos de recombinación (homóloga e ilegítima)

- Cualquier análisis filogenético y/o evolutivo de secuencias moleculares require de un **alineamiento** para poder comparar sitios homólogos entre las secuencias a estudiar. Para ello se escriben las secuencias en filas una sobre la otra, de modo que los sitios homólogos quedan alineados por columnas. Cada sitio o columna del alineamiento corresponde a un **caracter**, y los nt o aa que ocupan dichas posiciones representan los distintos **estados del caracter**



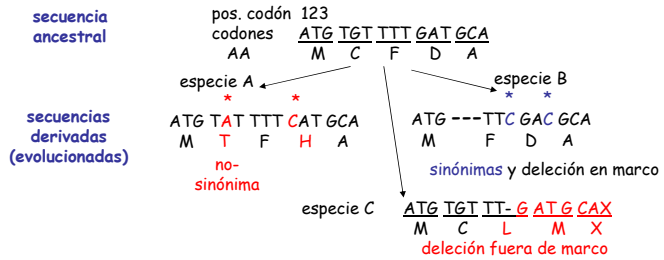
Homología entre secuencias de DNA y proteína: conceptos y terminología básica

- Cuando por eventos de inserción o deleción (**indeles**) las secuencias homólogas presentan distintas longitudes, es necesario introducir "**gaps**" en el alineamiento para mantener la correspondencia entre sitios homólogos situados antes y después de las regiones afectadas por indeles. Estas regiones se identifican mediante guiones (-). **Los indeles no se distribuyen aleatoriamente en las secuencias codificadoras**. Casi siempre aparecen ubicados entre dominios funcionales o estructurales, preferentemente en bucles (loops) que conectan a dichos dominios. Esto vale tanto para RNAs estructurales (tRNAs y rRNAs) como para proteínas. No suelen interrumpir el marco de lectura.



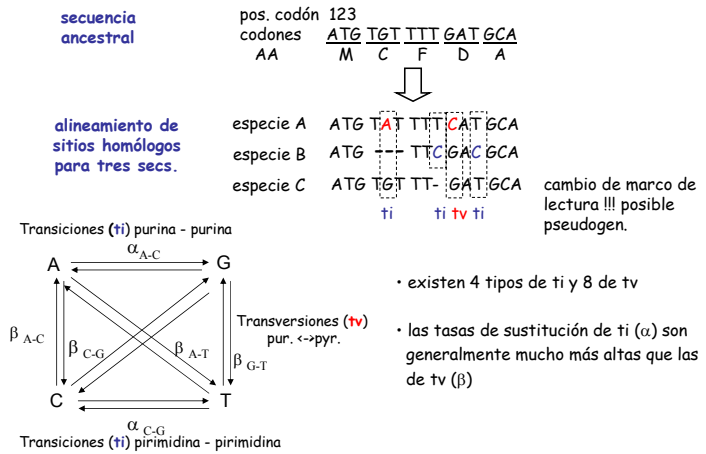
- A mayor **distancia genética** (evolutiva) entre un par de secuencias, mayor será el número de mutaciones acumuladas. Dependiendo del tiempo de separación de los linajes y la tasa evolutiva del locus, puede llegar a ser imposible alinear ciertas regiones debido a fenómenos de **saturación mutacional**. Las regiones de homología dudosa deben de ser excluídas de un análisis filogenético

Homología entre secuencias de DNA y proteína: tipos de mutaciones en secs. codificadoras de proteínas



- Todas las mutaciones en 2as posiciones resultan en sustituciones no sinónimas
- 96% de mutaciones en 1as posiciones resultan en sustituciones no sinónimas
- Casi todas las sustituciones sinónimas ocurren en las 3as posiciones
- las deleciones o inserciones en secs. codificadoras de aa suceden generalmente en múltiplos de tres nt; de no ser así se generan cambios de marco de lectura corriente abajo de la mutación, con frecuencia generando un pseudogen no funcional

Homología entre secuencias de DNA y proteína: alineamiento y tipos de mutaciones



- existen 4 tipos de ti y 8 de tv
- las tasas de sustitución de ti (α) son generalmente mucho más altas que las de tv (β)

Programación dinámica y la generación de alineamientos pareados (globales y locales)

• Estudiar el fundamento de los **algoritmos de PD** es un buen punto de arranque para entender lo que acontece dentro de software usado extensamente en biología computacional:

El corazón de programas como BLAST, FASTA, CLUSTALW, HMMER, GENSCAN, MFOLD y los de inferencia filogenética (PHYLIP, PAUP, MrBayes ...) emplean alguna forma de programación dinámica, con frecuencia **variantes heurísticas**

• **Alineamientos pareados: el problema visto desde la perspectiva biológica**

El supuesto básico es que si dos secuencias se parecen mucho a lo largo de sus secuencias es porque comparten un ancestro común : son homólogas.
Es decir, **inferimos la homología a partir de la similitud**.

Para cuantificar objetivamente el nivel de similitud necesitamos un sistema de puntuación (scoring scheme) que lo refleje adecuadamente, desde una perspectiva evolutiva

El objetivo es alinear las dos secuencias de tal manera que se maximice su similitud

Para ello necesitamos un algoritmo, ya que no es práctico evaluar todos los alineamientos posibles entre un par de secuencias dado el elevadísimo número de combinaciones $(2^{2N}/(2N)^{1/2}$. Así para dos secs. de 300 resíduos existen 10^{179} alns. posibles!!!

Los algoritmos de programación dinámica son adecuados para este trabajo

Programación dinámica y la generación de alineamientos pareados (globales y locales)

• Pares de secuencias pueden ser comparadas usando **alineamientos globales y locales**, dependiendo del objetivo de la comparación.

Un **alineamiento global** fuerza el alineamiento de ambas secuencias a lo largo de toda su longitud. Usamos aln. globales cuando estamos seguros de que la homología se extiende a lo largo de todas las secuencias a comparar. Este es el tipo de alineamientos que generan programas de alineamiento múltiple tales como clustal, T-Coffee o muscle.

(a)

```
P00001     1 MGDVEKGKKIFIMKCSQCHTVEKGGKHKTGPNLHGLFGRKTGQAPGYSYTAANKNK---GI   58
               D  KG+ +F     QC T  +  K+  GP L G+ GRK G A G++Y+  N N    G+
P00090     1 Q-DAAKGEAVF-----KQCMTCHRADKNMVGPALGGVVGRKAGTAAGFTYSPLNHNSGEAGL   56

P00001    59 IWGEDTLMEYLENPKKYIP--------------GTKMIFVGIKKKEERADLIAYLKKATNE   105
             +W ++ ++ YL +P  Y+                TKM F  +   ++R D+ AYL  AT +
P00090    57 VWTQENIIAYLPDPNAYLKKFLTDKGQADKATGSTKMTF-KLANDQQRKDVAAYL--ATLK   114
```

Alineamiento global óptimo del citocromo C humano (105 resíduos, SWISS-PROT acc. P00001) y citocromo C2 de Rhodopseudomonas palustris (114 resíduos, SWISS-PROT acc. P00090).

La matriz de puntuación o ponderación ("scoring matrix) empleada fue **BLOSUM62**, con **costo de gaps afines** de –(11 + k). La puntuación del alineamiento global es de 131, usando el algoritmo de **Needleman-Wunsch**.

Programación dinámica y la generación de alineamientos pareados (globales y locales)

Un **alineamiento local** sólo busca los segmentos con la puntuación más alta.
Se usa por ejemplo en el escrutinio de bases de datos de secuencias debido a que la homología entre pares de secuencias frecuentemente existe sólo a nivel de ciertos dominios, pero no a lo largo de toda la secuencia (**estructura modular de proteínas; genes discontínuos introns-exonesm; barajado de exones** ...).
BLAST y FASTA buscan alineamientos locales con alta puntuacion (**HSPs** ó high-scoring pairs)

(b)

```
P13569  1221 EGGNAILENISFSISPGQRVGLLGRTGSGKSTLLSAFLRLL-----NTEGEIQIDGVS 1273
             +   ++  +S ++  G+ + L+G +GSGKS   +A L +L      T GEI  DG
P33593    13 QAAQPLVHGVSLTLQRGRVLALVGGSGSGKSLTCAATLGILPAGVRQTAGEILADGKP   70

P13569  1274 WDSITL---------QQWRKAFGVIPQKVFIFSGTFRKNLDPYEQWSDQEIWKVADEV 1322
                L          Q  R AF  +              +  + + +   + K AD+
P33593    71 VSPCALRGIKIATIMQNPRSAFNPL------------HTMHTHARETCLALGKPADDA   116

P13569  1323 GLRSVIEQFP-GKLDFVLVDGGCVLSHGHKQLMCLARSVLSKAKILLLDEPSAHLDPV 1379
              L + IE         VL      +S G  Q M +A +VL ++  ++ DEP+  LD V
P33593   117 TLTAAIEAVGLENAARVLKLYPFEMSGGMLQRMMIAMAVLCESPFIIADEPTTDLDVV   174
```

Alineamiento local óptimo del regulador de conductancia transmembranal de fibrosis cística de humano (1480 resíduos, SWISS-PROT acc. P13569) y la proteína transportadora de Ni dependiente de ATP de E. coli (253 resíduos, SWISS-PROT acc. P33593).

La matriz de puntuación o ponderación ("scoring matrix) empleada fue **BLOSUM62**, con **costo de gaps afines** de –(11 + k). La puntuación del alineamiento local es de 89, usando el algoritmo de **Smith-Waterman**.

Programación dinámica y la generación de alineamientos pareados (globales y locales): dot plots y visualización de la similitud entre secuencias

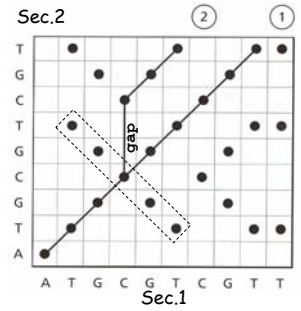
• las 2 secs. representan los dos ejes de la gráfica

• se pone un punto donde ambas coinciden

• la diagonal más larga representa la región de mayor identidad

• el camino 1 es el preferido al ser el más parsimonioso (implica menos cambios)

• la diagonal cruzada revela un palíndrome



alineamiento diagonal 1

```
secuencia 1:  ATGCGTCGTT
              ||||||| |||
secuencia 2:  ATGCGTCGT
```

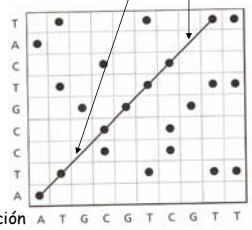
alineamiento diagonal 2

```
                  gap
secuencia 1:  ATG---CGTCGTT
              |||    |||
secuencia 2:  ATGCGTCGT
```

Programación dinámica y la generación de alineamientos pareados (globales y locales): dot plots y visualización de la similitud entre secuencias

```
secuencia 1:   ATGCGTCGTT
secuencia 3:   ATCCGTCAT
```

```
secuencia 1:   ATGCGTCGTT
               || |||| |
secuencia 3:   ATCCGTCAT
```

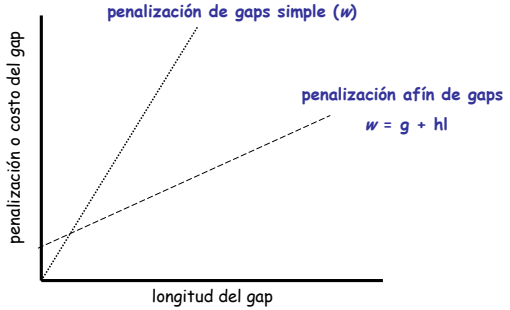


- la diagonal cruza celdas vacías, correspondientes a posiciones con distintos estados de caracter
- se pueden alinear dos secuencias aleatorias postulando una combinación de sustituciones y gaps
- se puede calcular el "costo" de un alineamiento contando el número de sustituciones (s) y gaps (g), o una función de ellos: p. ej.: $D = s + w$, donde w es un factor de penalización (FP) para la creación de gaps (**gap penalty**)
  donde para $w = 1$ abrir un gap cuesta igual que una sustitución
  $w = 2$ cuesta el doble un gap que una sustitución

  Se emplean valores bajos de w si pensamos abundaron indeles en la hist. evol. de las secs.
- **generalmente $w = g + hl$**, donde l es la longitud del gap, g es un FP de apertura del gap, y h es el FP para extender el gap. Estos son **FP afines**. La fórmula es muy flexible al permitir un control independiente del número y longitud (l) de los gaps mediante g y h

alineamientos pareados y factores de penalización afines para gaps

- Dado que **un sólo evento mutacional puede insertar o eliminar varios nucleótidos** de una secuencia, un indel largo no debe de ser penalizado mucho más que otro más corto ubicado en la misma región de un gen. De ahí el uso de **factores de penalización afines para gaps** (affine gap penalties or costs), que cobran una penalidad relativamente alta por abrir un gap y una penalidad más baja por cada posición sobre la que se extiende.
- La calidad de un alineamiento depende en gran medida de los **valores de apertura y extensión de gap** elegidos.



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

Un valor de puntuación es escogido para cada tipo de sustitución (par de resíduos o aln. de resíduo contra un gap). El set completo de estas puntuaciones conforman una matriz de ponderaciones o puntuaciones (**scoring matrix**), de dimensiones $S(i,j)$

Existen muchas definiciones del score de un alineamiento, pero la más común es simplemente la suma de scores o puntuaciones para cada par de letras alineadas y pares letra-gap, que conforman el alineamiento.

Así, para la matriz de sustitución siguiente y un w lineal de 5, calcula la puntuación del siguiente alineamiento

	-	A	G	C	T
A	10	-1	-3	-4	
G	-1	7	-5	-3	
C	-3	-5	9	0	
T	-4	-3	0	8	

AGACTAGTTAC
CGA---GACGT

Score = −3+7+10−3x5 +7−4+0−1+0 = 1

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

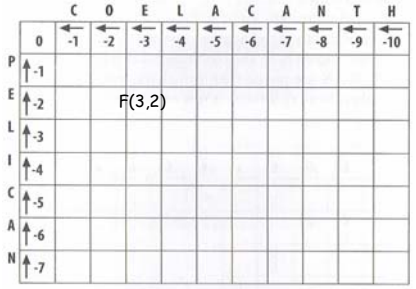
En realidad no se gurdan los caracteres en las celdas. Estas contienen dos valores: una puntuación (score) y un apuntador. El score se calcula a partir del esquema de puntuación o más generalmente, de una matriz de puntuaciones. El apuntador es un indicador de dirección (flecha) que apunta en una de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

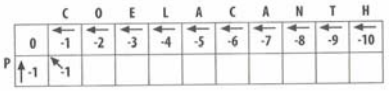
Programación dinámica: algoritmo de Needleman-Wunsch y alineamientos pareados globales

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

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



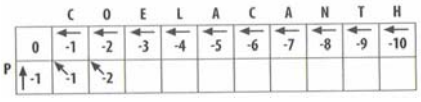
1. match score = 0 + (-1) = -1 ➔ es el score más alto y por tanto va a la celda
2. gap score horizontal = -1 + (-1) = -2
3. gap score vertical = -1 + (-1) = -2
4. la flecha apunta al 0 por ser el score vecino más alto

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      ó      CO
-P             P-
```

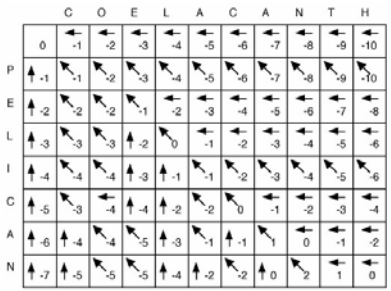
1. match score = -1 + (-1) = -2 ➔ es el score más alto y por tanto va a la celda
2. gap score horizontal = -1 + (-1) = -2
3. gap score vertical = -2 + (-1) = -3
4. la flecha puede apuntar al -1 de la diagonal u horizontal. Se toma una decisión arbitraria pero consistente si se vuelve a dar el caso (p.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

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

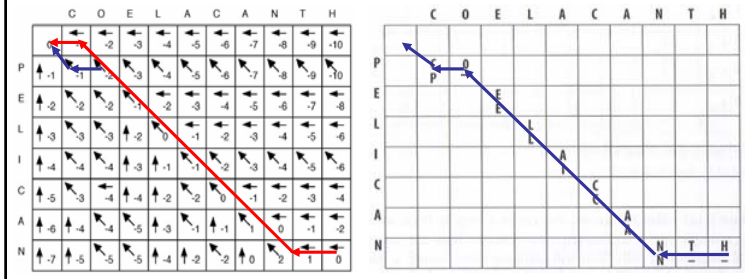


Programación dinámica: algoritmo de Needleman-Wunsch y alineamientos pareados globales

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

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

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

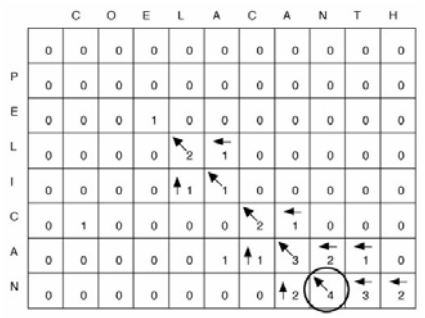
Smith TF, Waterman MS (1981) J. Mol. Biol 147(1);195-7

- 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

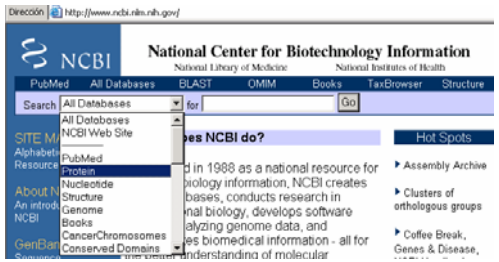
3°. Correr un alineamiento local con las proteínas P13569 y P33593 usando el programa WATER

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

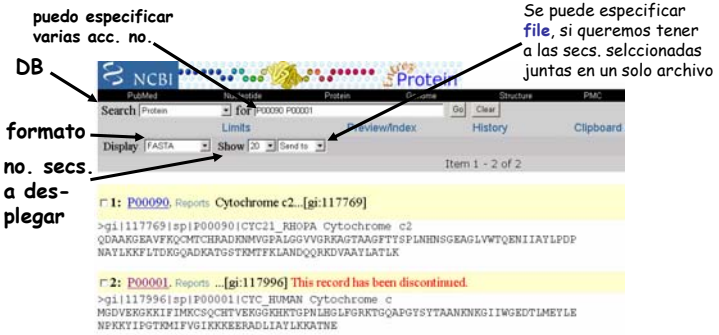
http://www.ncbi.nlm.nih.gov/



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 y Smith-Waterman 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.

Similitud entre pares de secuencias de AA

- El alineamiento de aa difiere del de nt en dos aspectos fundamentales:

  1.- Existen más "símbolos" en el alineamiento de aa (20) que de nt (4)

  2.- El alineamiento no consiste simplemente en alinear resíduos de tal manera que la mayor cantidad coincida, ya que hay que considerar los posibles **caminos mutacionales** mediante los cuales un aa es sutituído por otro

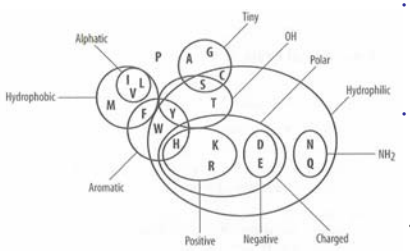
  Cys (UGU) → Tyr (UAU) 1 subst. en la 2a. pos del codón

  Cys (UGU) → Met (AUG) 3 subst. Una en cada posición del codón

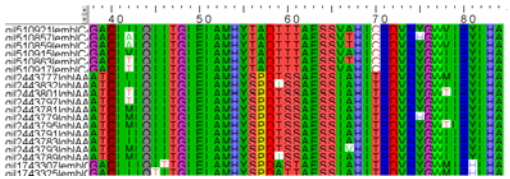
  Por lo tanto alinear Cys con Tyr es 3 veces menos costoso que alinearla con Met

- En el **alineamiento de nt** generalmente se valora un "match" como +1 y un "mismatch" como -3 (en NCBI BLAST), o como +5/-4 en WU-BLAST, es decir, los nt se consideran idénticos o distintos). Esto, unido a las penalizaciones de gap, define el costo de un alineamiento de nt
- Los **alineamientos de proteínas** se basan generalmente en una **matriz empírica de costo de sustitución**, derivada de la comparación de secuencias alineadas. Estas matrices empíricas reflejan someramente los caminos mutacionales.

Similitud entre pares de secuencias de AA

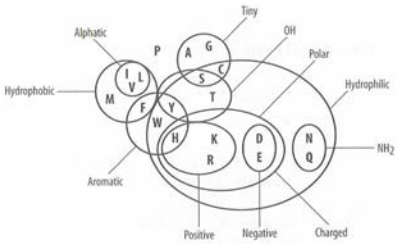


- Este diagrama muestra a los **aminoácidos** agrupados atendiendo a sus características químicas y físicas.
- Desde una perspectiva evolutiva **esperamos encontrar más sustituciones entre aas. similares que entre los menos relacionados.**
- Estos patrones puede observarse en alineamientos múltiples como el mostrado abajo

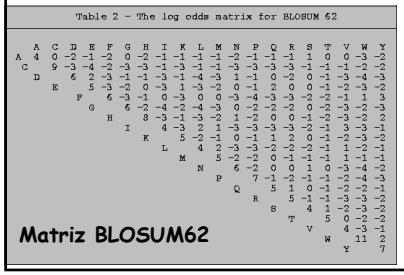


Segmento de un aln. múltiple de citocromos *C* de primates

Similitud entre pares de secuencias de AA



- **Las matrices empíricas de sustitución** entre AAs no reflejan necesariamente las relaciones químicas entre ellos. Se trata de una **definición púramente estadística basada en el análisis de frecuencias empíricas de sustituciones observadas en alineamientos de secs. con un grado de divergencia definido**



- Cada score de la matriz representa la tasa de sustitución esperada entre un par de AAs. Por tanto, los scores de los alineamientos pareados evaluados con estas matrices reflejan la distancia evolutiva existente entre las secuencias.

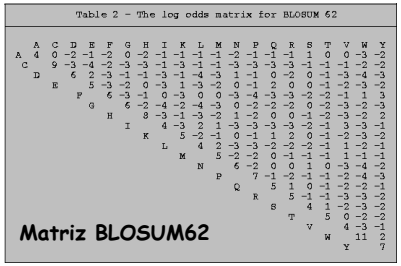
Es importante notar que los scores son evolutivamente simétricos al no conocerse la dirección del cambio evolutivo.

Similitud entre pares de secuencias de AA

- **Matrices de sustitución de AAs log-odds scores**

$$s(a,b) = (c) \log \frac{p_{ab}}{f_a f_b}$$

$s(a,b)$ = score del par a, b

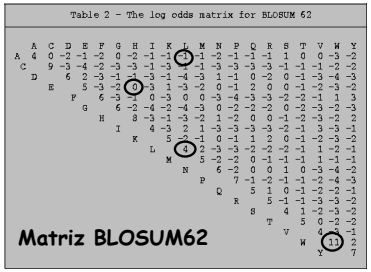


p_{ab} = verosimilitud de la hipótesis a testar; **frecuencia esperada o diana**, probabilidad con la que esperamos encontrar *a* y *b* apareados en un alineamiento múltiple

$f_a f_b$ = verosimilitud de la hipótesis nula; **frecuencia de fondo**, probabilidad con la que esperamos encontrar *a* y *b* en cualquier proteína. Refleja su abundancia o frecuencia

c = Factor de escalamiento usado para multiplicar los lod scores (números reales) antes de ser redondeados a números enteros, tal y como se observa en la matriz. Los valores enteros redondeados resultantes se conocen como "raw scores".

Similitud entre pares de secuencias de AA



Cálculo de "**scores crudos**"

$$s(a,b) = (c) \log \frac{p_{ab}}{f_a f_b}$$

- ¿Porqué difieren los valores entre diferentes sust. conservativas, por ej. L/L y W/W?

p_{LL} = 0.0371, p_{WW} = 0.0065

f_L = 0.099, f_W = 0.013

Las frecuencias de fondo juegan un papel muy importante. Cuanto más raro es un AA, menos frecuente será que se encuentre apareado consigo mismo por azar

- ¿Porqué se castiga más un apareamiento A/L (chico y alifático/alifático) con respecto a uno K/E (+/-)?

p_{AL} = 0.0044 f_L = 0.099, f_A = 0.074

p_{WW} = 0.0041 f_K = 0.058, f_E = 0.054

Similitud entre pares de secuencias de AA

- **Matrices de sustitución de AAs: ¿de dónde vienen los log-odds scores?**

- La **frecuencia diana** p_{ab} para un par de AAs corresponde a la **probabilidad esperada de encontrar *a*, *b* alineados en un alinemamiento de secuencias homólogas**

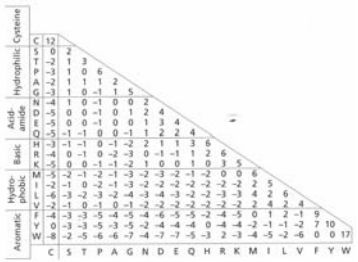
- Para **estimar** con la mayor precisión posible la frecuencia diana p_{ab} de cada par de AAs en una familia de secuencias homólogas hay que **analizar su distribución de frecuencia en muchos alineamientos confiables que difieren en el nivel de divergencia evolutiva** o distancia genética entre sus miembros.

- Cuanto más sepamos sobre la biología de las secuencias alineanadas, mejor podremos adecuar la estima de las frecuencias diana. Así p. ej. si alineamos prots de membrana, sus dominios transmembranales tendrán un fuerte sesgo hacia AAs hidrofóbicos, mientras que sus dominios extramembranales tendrán una mayor frecuencia relativa de AAs hidrofílicos. Se trata por tanto claramente de estimas empíricas y óptimas sólo para el caso analizado

- La distancia evolutiva entre las secuencias a analizar es una de las fuentes de información biológica más importantes para hacer una estima adecuada de p_{ab}. **Las frecuencias diana dependen fuertemente de la distancia evolutiva entre los pares de secs. analizadas.** Si divergieron recientemente, las frecuencias diana deben de ser ajustadas principalmente en base a resíduos idénticos. Cuanto más divergentes, la distribución de frecuencias diana debe de ser más plana. Por lo tanto las frecuencias diana se calculan en base a sets de aln. pareados confiables con distinto grado de divergencia. Se obtienen **series de matrices** correspondientes a estos distintos sets de alineamientos

Alineamiento pareado de proteínas: matrices de sustitución PAM

Derivación de una matriz de costo de sustitución PAM250 de M. O. Dayhoff (1978)



- Una medida de divergencia de secuencias de aa es PAM:
  1 PAM = 1 Percent Accepted Mutation
  (1 sustitución/100 resíduos)

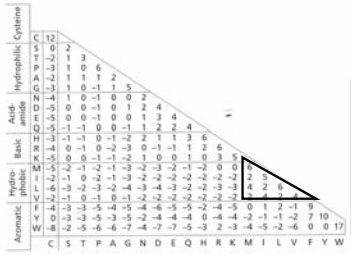
- Por tanto 2 secuencias que divergen en 1 PAM presentan un 99% de identidad

- Secuencias que divergen en sólo 1% de sus resíduos probablemente no hayan sufrido más que una sustitución/sitio

- Haciendo una recopilación de sustituciones entre secuencias con 1 PAM de divergencia, y corrigiendo para las abundancias relativas de los aa, se puede derivar una matriz de substitución PAM1

Alineamiento pareado de proteínas: matrices de costo PAM

Derivación de una matriz de costo de sustitución PAM250 de M. O. Dayhoff (1978)



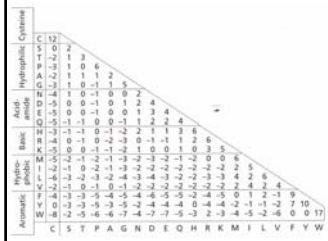
- Para producir una matriz apropiada para estimar similitud entre proteínas más divergentes se toman potencias de la matriz de sustitución PAM 1. Se trata de una aproximación muy teórica que no necesariamente refleja bien los patrones evolutivos de sustitución.

- El nivel PAM250, correspondiente a un nivel de identidad global del 20%, es el nivel de divergencia máximo para el que cabe esperar obtener un alineamiento plausible basado únicamente en el análisis de similitud entre las secuencias.

- La matriz da la relación de la frecuencia en al que los pares de aas son observados en comparaciones pareadas de proteínas existentes en bases de datos con respecto a aquellas esperadas por azar, expresadas como "log odds" (ver siguiente página). Lo aas intercambiados frecuentemente tienen una puntuación positiva, y aquellos que raramente reemplazan a otros tienen puntuación negativa. Nótese que los reemplazos ocurren más frecuentemente entre aas de propiedades fisico-químicas similares (ver como ejemplo los valores en el triángulo)

Alineamiento pareado de proteínas: matrices de costo PAM

Derivación de una matriz de costo de sustitución PAM250 de M. O. Dayhoff (1978)



- La existencia de reversiones (homoplasias) produce un **relentizamiento aparente** en tasas de sustitución debido a que una proporción creciente de posiciones variables de los alineamientos alcanzan el punto de **saturación mutacional**.

- Así la relación entre PAM score y % de identidad de secuencia es:

PAM	0	30	80	110	200	250
% identidad	100	75	50	40	25	20

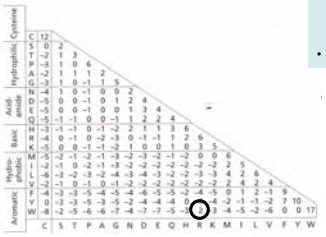
- Los valores de las tablas PAM vienen expresadas así:

$$\text{Valor mutación } i \leftrightarrow j = \log \frac{\text{tasa observada } i \leftrightarrow j}{\text{tasa esperada en base a la freq. de aa}}$$

- Este valor se multiplica X10 para evitar decimales

Alineamiento pareado de proteínas: matrices de costo PAM

Derivación de una matriz de costo de sustitución PAM250 de M. O. Dayhoff (1978)



- Los valores de las tablas PAM vienen expresadas así:

Valor mutación i <-> j = log $\frac{\text{tasa observada i <--> j}}{\text{tasa esperada en base a la freq. de aa}}$

- Este valor se multiplica X10 para evitar decimales

Así un valor +2 (p.ej. W <-> R) implica que la mutación acontece 1.6 veces más frecuentemente que lo esperado por azar. El valor +2 corresponde a 0.2 debido al factor de escalamiento. El valor 0.2 es el log10 del valor de expectación relativa de la mutación. Así el valor de expectación es $10^{0.2}$=1.6

- La probabilidad de dos eventos mutacionales independientes es el producto de sus probabilidades. Al usar logs, se tienen puntuaciones (scores) que se suman en vez ser multiplicadas, lo que es ventajoso desde una perspectiva computacional

Alineamiento pareado de proteínas: matrices de costo BLOSUM

Matrices BLOSUM de sustitución de aa
Henikoff, S., Henikoff, J. G., and Pietrokovski, S. 1999. Blocks+: a non-redundant database of protein alignment blocks derived from multiple compilations. Bioinformatics **15**: 471-479.

- Desarrollada por S. Henikoff y J. G. Henikoff para obtener una matriz más robusta que las PAM en la identificación de homólogos distantes, particularmente cuando contienen una proporción significativa de aas hidrofóbicos
- Las matrices **BLOSUM** están basadas en la base de datos **BLOCKS+** de proteínas alineadas; **BLO**cks **SU**bstitution **M**atrix (http://blocks.fhcrc.org). Son matrices empíricas.
- Las series de matrices **BLOSUM** se derivaron de alineamientos sin indeles (**BLOCKS**) de proteínas **considerando sólo pares de alineamientos que no divergieran más de un umbral determinado**, por ej. un **mínimo de 62 % de identidad**, para calcular las frecuencias diana o esperadas de la **matriz BLOSUM62**. Para estos alns. se calcula la razón entre el número de pares de aa observados en cada posición y el número de pares esperados de las frequencias globales de los aas, expresando los resultados como $\log_{10} X \lambda$
- Para evitar sesgos en las matrices por sobrerepresentación de secuencias muy similares, se reemplazaron aquellas con similitud > a un umbral dado por un solo representante o por un promedio ponderado (BLOCKS+).
- **La matriz BLOSUM62 es la actualmente favorecida para la mayoría de las aplicaciones por su buen rendimiento empírico y ha reemplazado a las matrices de Dayhoff (PAM)**

Alineamiento de proteínas: selección de matrices de ponderación - consejos prácticos

- Las **matrices PAM** fueron derivadas de las secuencias de proteínas disponibles a finales de los 60s y ppios. de los 70s. Era una base de datos muy reducida y estaba **sesgada a proteínas chicas, globulares e hidrofílicas** ! Al carecer de suficientes homólogos con diversos niveles de divergencia evolutiva tuvieron que emplear supuestos teóricos (extrapolación) para obtener las matrices de sustitución para prots. más distantes (mediante exponenciación)

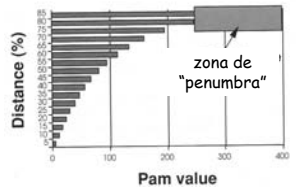
- las matrices PAM son una pobre elección para alinear (o buscar en las bases de datos) proteínas con dominios hidrofóbicos (p. ej. dominios transmembrana)

- Qué matriz escoger en función del nivel de divergencia esperada **(potencial de mira retrospectiva en tiempo evolutivo)**

% identidad	PAM	BLOSUM	mira retrospectiva en tiempo evolutivo
20- 50 %	250	45	homólogos en la zona de penumbra
50- 75 %	250	62	ortólogos y parálogos en superfamilias[1]
75- 90 %	160	80	ortólogos y parálogos en familias[2]
90- 99 %	40	90	ortólogos muy cercanos

[1]Superfamilias de proteínas contienen diversas familias de proteínas con ≥ 30% identidad entre ellas
[2]Familias de proteínas contienen secuencias con ≥ 85% identidad entre ellas
Estas definiciones fueron acuñadas por Dayhoff et al. (1978)

Alineamiento de proteínas: selección de matrices de ponderación - consejos prácticos para la identificación de homólogos



- A medida que el nivel de divergencia entre pares de proteínas alcanza el valor de PAM250 (~ **20% identidad**), comienza a ser dudosa su relación de homología, pudiendo tratarse de secuencias que presentan cierto grado de similitud por azar, en base a composiciones de AAs similares en ambas secuencias !!!

- Al entrar en esta **zona de penumbra**, es esencial considerar información adicional, particularmente motivos estructurales, para validar o descartar una posible relación de homología

Distancias observadas vs. evolutivas (PAM) entre prots.

Diferencia % obs.	Dist. evol. PAM	
1	1	
5	5	
10	11	
15	17	
20	23	
30	38	
40	56	
50	80	
60	112	
70	159	
80	246	
85	328	← z. penumbra

- A medida que el nivel de divergencia evolutiva entre pares de proteínas incrementa (distancias PAM) disminuye el número de diferencias observadas, debido a fenómenos de **reversión** (**homoplasia**). Por tanto, si no se cuenta con evidencia estructural, el análisis filogenético de proteínas debe restringirse a aquellas con ≥ 20% de identidad. **Los alns. tampoco son confiables**

Alineamiento de proteínas: selección de matrices de ponderación - consejos prácticos para la identificación de homólogos

- Clasificación de familias de proteínas atendiendo a su nivel de antigüedad evolutiva I

1. Proteínas antiguas

A) "primeras ediciones": básicamente enzimas del metabolismo central y proteínas involucradas en los procesos de procesamiento de la información genética
Ejs. **trifosfato isomerasa** (TPI), **glutamato deshidrogrenasa**, **aminoacyl-tRNA sintetasas**, **proteínas ribosomales** ...

B) "segundas ediciones": homólogos en eucariotes y procariontes, pero con funciones diferenciadas.
Ej: **glutation reductasa humana** y la **reductasa de Hg de *Pseudomonas*** (31% I a lo largo de 438 aa, ($E<10^{-32}$)

2. Proteínas de la "edad media"

homólogos en eucariotes pero ausentes en procariontes. Ej: **actina** humana y la de levadura 88% de I a lo largo de 375 aa, ($E<10^{-145}$); otras actinas de levaduras sólo 26% de I a lo largo de 489 aa, ($E<10^{-14}$)

Alineamiento de proteínas: selección de matrices de ponderación - consejos prácticos para la identificación de homólogos

- Clasificación de familias de proteínas atendiendo a su nivel de antigüedad evolutiva (II)

3. Proteínas "modernas"

A) de invención reciente: presentes en plantas o animales pero no en los dos reinos. No presentes en procariontes. Ej. **colágeno**

B) de invención muy reciente. Por ej. proteínas presentes sólo en vertebrados, tal como la **albúmina del plasma sanguíneo**

C) mosaicos recientes: proteínas modernas resultantes del barajado de exones (exon-shuffling) como el **receptor de LDL** o **activador de plasminógeno**

Alineamiento de proteínas: selección de matrices de ponderación - consejos prácticos para la identificación de homólogos

1. Para identificar homólogos lejanos de genes codificadores de proteínas, **comparar siempre las secuencias de los productos génicos**. Sólo en ellos quedan reflejadas las constricciones evolutivas que les permiten mantener plegamientos y funcionalidades a lo largo de grandes distancias evolutivas. De ahí la importancia de **incorporar análisis estructurales para la determinación de homología entre secuencias distantes**

2. Las secuencias homólogas comparten un ancestro común y por tanto un **plegado común**. Dependiendo de la distancia evolutiva y el camino de divergencia, dos o más homólogos pueden compartir muy pocos resíduos estrictamente conservados a nivel de la secuencia primaria. Pero, **si se ha podido inferir homología significativa entre A y B, entre B y C y entre C y D, entonces A y D tienen que ser también homólogos entre ellos**, aún cuando presenten < 20% de identidad

Cuantificación y análisis estadístico de la similitud entre un par de secuencias

- **Conceptos básicos de teoría de la información**

- **INFORMACIÓN** = decremento en el nivel de incertidumbre

- cualitativamente esperamos mayor contenido de información en un **vocabulario rico** que en uno pobre y en **respuestas sorprendentes** que esperadas.
Por tanto la información o sorpresividad de una respuesta es inv. prop. a su probabilidad

- cuantitativamente la **información** (H) o **entropía** asociada a un valor de **propabilidad** (p) viene expresada por la siguiente expresión:

$$H(p) = \log_2 1/p = -\log_2 p$$

- valores convertidos a $\log_2$ se les asigna la unidad **bit** (binary digit), mientras que los que son convertidos a **log** en base **e** tienen por unidad los **nats** (natural digits).

- Se describe frecuentemente a la información como un **mensaje** de **símbolos** emitido por una **fuente**. Los símbolos presentan una **distribución de frecuencia**

- Si dicha distribución es plana y existen n símbolos, la p para cada símbolo es $1/n$
La infromación de cada uno de estos símbolos es su **entropía** = $\log_2 (1/n)$

Cuantificación y análisis estadístico de la similitud entre un par de secuencias

• **Conceptos básicos de teoría de la información**

- Si la distribución de frecuencias no es equiprobable, para calcular la entropía de cada símbolo hay que ponderarla por su *p* (frecuencia) de ocurrencia.

$$H = -\sum_{i}^{n} p_i \log_2 p_i$$ **Indice de entropía de Shannon**

Ej. 1: para una moneda estándar su entropía es de 1 bit

- ((0.5)(-1) + (0.5)(-1)) = 1 bit

Ej. 2: para una moneda trucada en la que p águila es de 0.75 su entropía es de 0.51 bits

- ((0.75)(-0.415) + (0.25)(-2)) = 0.81 bits

Ej. 3: La entropía de una fuente aleatoria de secuencia de DNA es de 2 bits

- ((0.25)(-2) + (0.25)(-2) + (0.25)(-2) + (0.25)(-2)) = 2 bits

Ej. 4: una fuente de DNA que emite 90% de A ó T y 10% de G ó C es de 1.47 bits

- (2(0.45)(-1.15) + 2(0.05)(-4.32)) = 1.47 bits

Cuantificación y análisis estadístico de la similitud entre un par de secuencias

• Un script de Perl que calcula la entropía de un archivo usando el índice de Shannon

```
#!/usr/bin/perl -w
use strict;

# Calculadora de entropía de Shannon
my %Count;
my $total =0;

while (<>) {
    foreach my $char (split(//,$_)){
        $Count{$char}++;
        $total++;
    }
}

my $H =0;
foreach my $char (keys %Count){
    my $p =$Count{$char}/$total;
    $H +=$p *log($p);
}

$H = -$H/log(2);

print "H = $H bits \n";
```

Explica lo que hace este script

Cuantificación y análisis estadístico de la similitud entre un par de secuencias

• Un script de Perl que calcula la entropía de un archivo

```
#!/usr/bin/perl -w
# Calculadora de entropía de Shannon : Shannons_H-calculator.pl

# uso: perl Shannons_H-calculator.pl <nombrearchivo>; ó ./miscript <nombrearchivo>
use strict;      # activamos la directiva "strict" que nos obliga a declarar vars.

# 0) Inicializamos y declarmamos variables
my %Count;       # declaramos un hash que almacenará las cuentas de cada símbolo
my $total =0;    # inicializamos un contador de símbolos totales

# 1) Construímos la estructura de datos: un hash o arreglo asociativo.
while (<>){       # leemos líneas del archivo de entrada
    foreach my $char (split(//,$_)){ # obtenemos caracteres indiv. de cada palabra
        $Count{$char}++;             # autoincrementamos el valor de cada caracter
        $total++;                    # autoincrementamos el contador
    }
}
# 2) Iteramos sobre los valores del hash para hacer el cálculo de H
my $H =0;        # inicializamos la variable H (entropía)
foreach my $char (keys %Count){    # iteramos sobre el hash de caracteres
    my $p =$Count{$char}/$total;   # probabilidad de cada caracter o símbolo
    $H +=$p *log($p);              # Cálculo de la entropía: sumatoria de p *log(p)
}
$H = -$H/log(2);        # negativizamos la suma (H), convertimos base "e" a base 2

print "H = $H bits \n";   # imprimimos el resultado a STDOUT (salida a pantalla)
```

Estadísticos de Karlin-Altschul de similitud entre secuencias: frecuencias diana, lambda y entropía relativa

Los atributos más importantes de una matriz de sustitución son sus **frecuencias esperadas o diana** implícitas para cada par de aa en sus respectivos scores crudos. Estas frecuencias esperadas **representan el modelo evolutivo subyacente**.
Los scores que han sido re-escalados y redondeados (scores representados en la matriz) son los **scores crudos** $s_{a,b}$. Para convertirlos a un **score normalizado** (log-odd score original) tenemos que mutiplicarlos por λ, una constante específica para cada matriz.
λ es aprox. igual al inverso del factor de escalamiento (*c*).

$$s(a,b) = \frac{1}{\lambda} \log \frac{p_{ab}}{f_a f_b} \qquad p_{ab} = f_a f_b \, e^{\lambda s_{ab}}$$ λs_{ab} = **score normalizado**

por tanto, para despejar λ necesitamos $f_a f_b$ y encontrar el valor de λ para el que la suma de las frecuencias diana implícitas valga 1.

$$\sum_{a=1}^{n}\sum_{b=1}^{a} p_{ab} = \sum_{a=1}^{n}\sum_{b=1}^{a} f_a f_b \, e^{\lambda s_{ab}} = 1$$

Una vez calculada λ, se usa para calcular el **valor de expectación** (*E*) de cada **HSP** (**High Scoring Pair**) en el reporte de una búsqueda **BLAST**

Dado que las $f_a f_b$ de los resíduos de algunas proteínas difieren mucho de las frecuencias de resíduos empleadas para calcular las matrices PAM y BLOSUM, versiones recientes de BLASTP y PSI-BLAST incorporan una "**composition-based** λ" que es "**hit-específica**"

Estadísticos de Karlin-Altschul de similitud entre secuencias: frecuencias diana, lambda y entropía relativa

$$\sum_{a=1}^{n}\sum^{a} p_{ab} = \sum_{a=1}^{n}\sum^{a} f_a f_b \, e^{\lambda s_{ab}} = 1$$

El valor de λ que permite resolver esta ecuación existe siempre y cuando la matriz de sustitución cumpla dos propiedades:

1.- ha de presentar al menos un score positivo

2.- el score esperado para alineamientos pareados de secuencias aleatorias ha de ser negativo

Ambas condiciones las cumplen las matrices generadas por cálculo de log-odds

Estadísticos de Karlin-Altschul de similitud entre secuencias: frecuencias diana, lambda y entropía relativa

· **Score esperado (*E*) y Entropía relativa (*H*)**

El **score esperado de una matriz de sustitución** es la suma de sus scores crudos ponderados por su frecuencia de ocurrencia. Este score esperado ha de ser **siempre negativo**.

$$E = \sum_{a=1}^{20}\sum^{a} f_a f_b \, s_{ab}$$

La **entropía relativa de una matriz de sustitución** resume su comportamiento general de manera conveniente. Se calcula a partir de los scores normalizados. ***H*** **es el número promedio de bits (o nats) por resíduo en un alineamiento y es siempre positivo.**

$$H = -\sum_{a=1}^{20}\sum^{a} p_{ab} \, \lambda \, s_{ab}$$

Así por ej. *H* de PAM1 es > *H* de PAM120, esta última contiene menos información por ser menos específica. De igual manera BLOSUM80 contiene más información que BLOSUM62. Para calcular las equivalencias entre matrices PAM y BLOSUM se comparan a nivel de sus *H*s.

H de PAM250 ≈ BLOSUM45; *H* de PAM180 ≈ BLOSUM80; *H* de PAM180 ≈ BLOSUM62

Estadísticos de Karlin-Altschul para alineamientos locales

Karlin, S., and Altschul, S. F. 1990. Methods for assessing the statistical significance of molecular sequence features by using general scoring schemes. Proc Natl Acad Sci U S A **87**: 2264-268.

Los estadísticos de Karlin-Altschul asumen 5 supuestos:

1. Un score positivo ha de ser posible
2. El score esperado ha de ser negativo
3. Los resíduos de una secuencia son independientes y distribuídos idénticamente
4. Las secuencias son infinitamente largas
5. Los alineamientos no contienen gaps

Los primeros dos supuestos los cumple cualquier matriz estimada a partir de datos reales. Los tres supuestos finales son problemáticos. Se han solucionado en trabajos posteriores.

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

Esta ecuación indica que el **número de alineamientos esperados por azar** (*E*) durante una búsqueda de similitud en una base de datos de secuencias está en función de:
el tamaño del espacio de búsqueda (*m*, *n*), el score normalizado (λ*S*) del HSP y una constante de valor pequeño (*k*)

***E* Describe el ruido de fondo por azar presente en matches de dos secs.**

m = número de símbolos en la secuencia problema
n = número de símbolos en la base de datos
k ≈ 0.1 constante de ajuste para considerar HSPs altamente correlacionados

BLAST: Basic Local Alignment Search Tool

Altschul, S. F., Gish, W., Miller, W., Myers, E. W., and Lipman, D. J. 1990. Basic local alignment search tool. J Mol Biol **215**: 403-410.

Altschul, S. F., Madden, T. L., Schaffer, A. A., Zhang, J., Zhang, Z., Miller, W., and Lipman, D. J. 1997. Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. Nucleic Acids Res **25**: 3389-402.

Schaffer, A. A., Aravind, L., Madden, T. L., Shavirin, S., Spouge, J. L., Wolf, Y. I., Koonin, E. V., and Altschul, S. F. 2001. Improving the accuracy of PSI-BLAST protein database searches with composition-based statistics and other refinements. Nucleic Acids Res **29**: 2994-3005.

BLAST: Basic Local Alignment Search Tool

BLAST consta de una **familia de programas**. Los 5 ppales son:

BLASTN (nt-nt), **BLASTP** (p-p), **BLASTX** (translated nt-p),
TBLASTN (p-translated nt), usado en mapeo de prots contra DNA genómico
TBLASTX (translated nt - translated nt) usado en la predicción de genes

y **variantes de BLASTP** como **PSI-** y **PHI-BLAST**

BLAST: Basic Local Alignment Search Tool

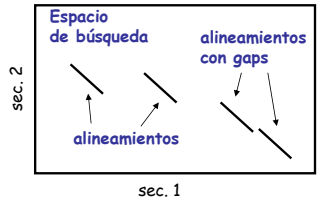
Gish, W, and DJ States 1993. Identification of protein coding regions by database similarity search. Nature Genetics **3**:266-72.

WU-BLAST en línea por ej. en:
http://www.ebi.ac.uk/blast2/

BLAST: Basic Local Alignment Search Tool

• **El algoritmo BLAST**

El **espacio de búsqueda** entre 2 secs. puede ser visualizado como una gráfica con una sec. en cada eje. Sobre esta gráfica podemos visualizar **alineamientos** como una secuencia de pares de letras con o sin gaps. Score = sumatoria de scores individuales p_{ab} – costo gaps.



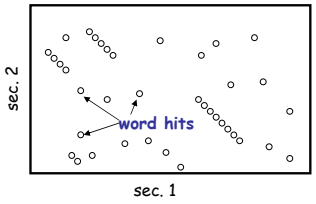
BLAST reporta todos los alns. pareados (HSPs) estadísticamente significativos encontrados en su búsqueda heurística del espacio de búsqueda. Hay que entender que en las búsquedas BLAST siempre hay que hacer un **compromiso entre velocidad** y **sensibilidad**. La velocidad se gana al no explorar toda la matriz, perdiéndose sensibilidad (*vs.* SM)

El **algoritmo heurístico de BLAST** sigue tres niveles de reglas para refinar secuencialmente HSPs (High Scoring Pairs) potenciales: **ensemillado**, **extensión** y **evaluación**. Estos pasos **conforman una estrategia de refinamiento secuencial que le permite a BLAST muestrear todo el espacio de búsqueda sin perder tiempo en regiones de escasa similitud**

BLAST: Basic Local Alignment Search Tool

• **Ensemillado**

BLAST asume que los alineamientos significativos contienen "**palabras**" en común (serie de letras). BLAST primero determina la localización de todas las palabras comunes ("**word hits**"). Sólo las regiones que contienen word hits serán usados como semillas de alineamientos. Así se reduce mucho el espacio a explorar.



MPRDG { MPR, PRD, RDG } secuencia y palabras de 3 letras

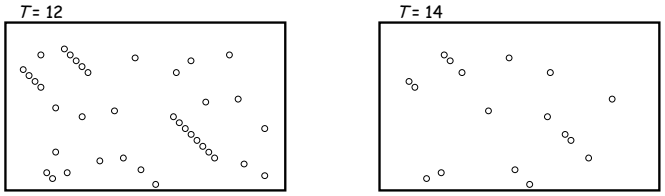
BLAST usa el concepto de **vecindad** para definir un *word hit*. Esta contiene a la palabra misma y todas las demás cuyo score sea al menos tan grande como *T* cuando se compara con la matriz de pesado. *T* corresponde a un threshold (umbral) mínimo de score que han de tener las palabras encontradas. Vecinos aceptados de RDG serían:

Palabra	Score (Blosum62)
RDG	17
KGD	14
QGD	13
RGE	13
EGD	12
...	

BLAST: Basic Local Alignment Search Tool

· **Ensemillado**

El valor adecuado de *T* depende de los valores en la tabla de sustitución empleada, como del balance deseado entre velocidad y sensibilidad. A valores más altos de *T*, menos palabras son encontradas, reduciendo el espacio de búsqueda. Ello hace las búsquedas más rápidas, a costa de incrementar el riesgo de perder algún alineamiento significativo.

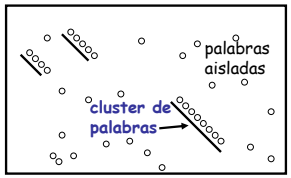


El **tamaño de palabra** *W* es otro parámetro que controla el número de word hits. *W*=1 producirá más hits que *W* = 5. Cuanto más chico sea *W* más sensible y lenta la búsqueda.
La interrelación entre *W*, *T* y la matriz de sustitución empleada es crítica, y su selección juiciosa es la mejor manera de controlar el balance entre velocidad y sensibilidad de BLAST

BLAST: Basic Local Alignment Search Tool

· **Ensemillado**

Las palabras tienden a agruparse en clusters en algunas regiones del espacio. BLAST usa el **two-hit algorithm** para seleccionar regiones con al menos dos palabras agrupadas dentro de una distancia definida sobre la diagonal. De esta manera **se eliminan palabras sin significancia, que carecen de vecinos**. Cuanto más grande la distancia impuesta al algoritmo (*A*), más palabras aisladas serán ignoradas, reduciéndose consecuentemente el espacio de búsqueda, incrementándose la velocidad a costa de perder sensibilidad.



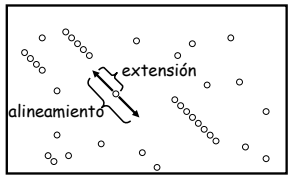
Detalles de implementación:
En **NCBI-BLASTN** las semillas son siempre palabras idénticas. *T* no es usado. Para hacer BLASTN más rápido se incrementa *W*, par hacerlo más sensible se disminuye *W*. El valor min. de *W* = 7. El algoritmo de two-hit tampoco es usado por BLASTN ya que hits de palabras largas idénticas son raros.

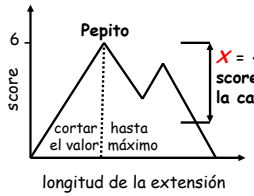
BLASTP (y otros programas basados en aa) usan valores de *W* de 2 ó 3. Para hacer las búsquedas más rápidas *W* = 3 y *T* = 999, que elimina todas las palabras vecinas. La distancia (*A*) entre vecinos del algoritmo two-hit es por defecto = 40 aas.
Las palabras que ocurren con una frecuencia significativamente mayor que la esperada por azar (FFF) corresponden frecuentemente a **regiones de baja complejidad** (rbc) que generalmente **son enmascaradas**. El uso de "**soft masking**" evita el ensemillado en rbc

BLAST: Basic Local Alignment Search Tool

· **Extensión**

Una vez que el espacio de búsqueda ha sido ensemillado, pueden generarse alineamientos pareados a partir de semillas individuales. La extensión acontece en ambas direcciones.





En el algoritmo de Smith-Waterman los puntos terminales de un aln. local son determinados después de haber evaluado todo el espacio de búsqueda. **BLAST**, al ser un algoritmo heurístico, tiene un mecanismo para no tener que explorar todo el espacio de búsqueda y **sólo extiende una semilla hasta un determinado punto**. Para ello se requiere de una **variable** *X*, que **representa cuánto se permite caer al score del alineamiento después de haber pasado por un máximo**. El algoritmo lleva la cuenta de los scores del alineamiento y de caída en base a la matriz de sustitución y de penalización de gaps

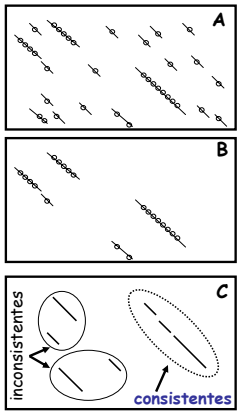
Ej. del control de extensión usando +1/-1 para match y mismatch respect., *X* = **4**, (no gaps)

```
Pepito Pérez se fue a pescar al lago
Pepito López no vio a Arturo en casa
123456 54345 43 210  1 0 ... <- score aln.
000000 12321  23 456 5 6 ... <- score de caída
```

BLAST: Basic Local Alignment Search Tool

· **Evaluación**

Una vez extendidas las semillas, los **alns.** resultantes son evaluados para determinar si son **estadísticamente significativos**. Los que lo son se denominan **HSPs** (**high scoring pairs**)



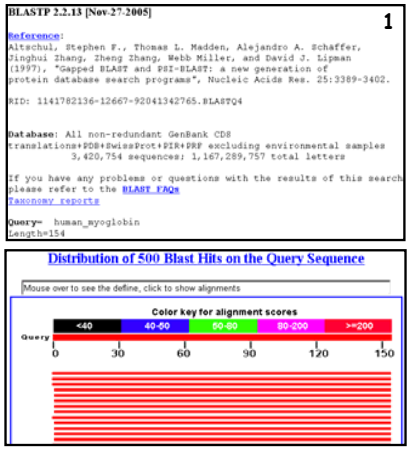
Determinar la significancia de múltiples HSPs no es tan sencillo como sumar los scores de todos los alns. involucrados, ya que muchos corresponden a extensiones de palabras fortuitas, por lo que no todos los grupos de HSPs tienen sentido. Se define así un **umbral de alineamiento** (**aln. threshold** *AT*), basado en los scores de los alns. y que no considera por tanto el tamaño de la base de datos (BD). Cuanto más alto, menos alns. son considerados (Figs. *A* y B).

Idealmente la relación entre los HSPs debería de ser lo más parecida posible a alns. sin gaps globales, es decir, segir las diagonales por la mayor distancia posible y no solaparse.

Grupos de HSPs que se comportan de esta manera se denominan **grupos consistentes de HSPs** (Fig. *C*). Para identificarlos, el algoritmo determina las coordenadas de todos los HSPs para cuantificar el solape. Este cálculo es cuadrático. Una vez organizados en grupos consistentes, se calcula un "**final threshold**" para cada grupo que considera todo el espacio de búsqueda (tamaño de la BD). **BLAST reporta todos los que están por encima del** *E* **value de corte**

BLAST: Basic Local Alignment Search Tool

- Anatomía de un reporte de NCBI-BLAST estándar



1.- **Encabezado**. Indica el programa de BLAST y su versión, con la fecha

Request ID

Indica la **BD** sobre la que se hizo la búsqueda, junto con el no. de secs contenida en ella y el no. de caracteres

Indica cual fue la query y su longitud

2.- **Resumen gráfico de distribución de hits con respecto a la query**.

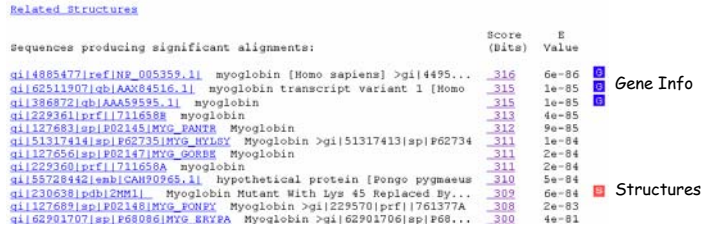
escala de color que indica el score de los HSPs

Las barras indican la distribución de los HSPs (coordenadas) con respecto a la secuencia problema (query), indicando en una escala de color el score de los alns. medidos en bits

BLAST: Basic Local Alignment Search Tool

- Anatomía de un reporte de NCBI-BLAST estándar

3. **Resúmenes de 1 linea**. Indican el nombre de la sec. junto con el score más alto y E value más bajo encontrado para un HSP o grupo de HSPs



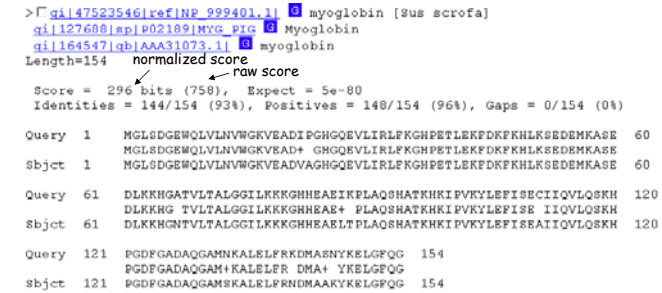
Gene Info

Structures

BLAST: Basic Local Alignment Search Tool

- Anatomía de un reporte de NCBI-BLAST estándar

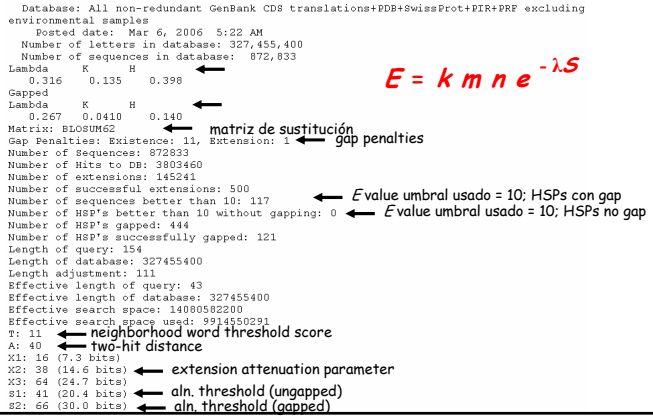
4. **Alineamientos**. Representan la parte más voluminosa del reporte. Además de la información estadística, indica las coordenadas de inicio y fin de las secuencias query y subject. Si la búsqueda involucra secuencias de DNA, también se indica direccionalidad de las hebras Q/S (plus/plus; plus/minus).



BLAST: Basic Local Alignment Search Tool

- Anatomía de un reporte de NCBI-BLAST estándar

5. **Pie de página**. Reporta los parámetros de búsqueda y varios estadísticos. Los más importantes son: **DB**, *T*, *E* y la **matriz de sustitución** o esquema de puntuación (match/missmatch) y **gap penalties** empleados



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

BLAST: Basic Local Alignment Search Tool

- **Anatomía de un reporte de NCBI-BLAST estándar**

6. **Cladogramas o árboles de NJ o ME.** Navegar por los hits en forma de árboles

BLAST: Basic Local Alignment Search Tool

- **RESUMEN de gapped-BLAST**

- BLAST es un progrma para búsqueda de secuencias similares a una sec. problema en bases de datos. BLAST puede ser usado en línea o localmente.

- Existen **diversos programas BLAST** para comparar todas las combinaciones posibles de secs. problema (aa y nt) con nt o aa DBs. (**BLASTN, BLASTP, BLASTX, TBLASTN, TBLASTX**) además de variantes de éstos que buscan similitudes en diversas DBs

- BLAST es una **versión heurística del algoritmo de Smith-Waterman** que encuentra matches locales cortos (**palabras**) que intenta extender en forma de alineamientos pareados

- El nuevo algoritmo **gapped-BLASTP** requiere al menos de dos palabras o hits no solapados con un score de al menos *T*, ubicados a una distancia máxima *A* el uno del otro, para invocar una extensión del segundo hit. Si el **HSP** generado tiene un score normalizado con un valor de al menos *Su* (**normalized ungapped score**) bits, se dispara una extensión con gap

- BLAST reporta además información relativa a la significancia estadística de los HSPs encontrados. El estadístico fundamental es el **valor de expectancia *E*** (***E*-value**), que indica la tasa de falsos positivos que cabe encontrar, dada la longitud de la secuencia problema, el tamaño de la base de datos exprolada, y el score normalizado del HSP, tal y como indica la **ecuación de Karlin-Altschul**

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

- Si bien no existe una teoría estadística para evaluar explícitamente la significancia de alns. con gaps (no se puede estimar λ) éstas pueden obtenerse a partir de simulaciones *in silico*

Identificación de homólogos lejanos mediante PSI-BLAST

La búsquedas de secuencias distantes en bases de datos mediante **matrices de ponderación sitio específicas** (también conocidas como **perfiles** o **motivos**) son generalmente más adecuadas para la identificación de homólogos con bajo nivel de identidad que el BLASTP estándar

PSI-BLAST (Position-Specific Iterated BLAST) es una modificación de BLASTP que permite la búsqueda de homólogos mediante **perfiles generados automáticamente a partir de alineamientos múltiples derivados de los HSPs encontrados por BLASTP.**

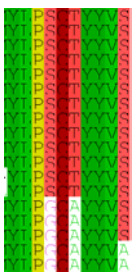
- Pasos que sigue el algoritmo de PSI-BLAST

1. Búsqueda de homólogos de una sec. problema mediante BLASTP
2. Construcción de un aln. múltiple a partir de los HSPs y construcción de un perfil
3. El programa compara el perfil construido con la base de datos
4. PSI-BLAST determina la significancia estadística de los alns. locales encontrados
5. PSI-BLAST puede repetir o iterar los pasos a partir del 2. para construir perfiles cada vez más específicos con las secuencias nuevas encontradas en cada iteración hasta llegar a la convergencia

Identificación de homólogos lejanos mediante PSI-BLAST

- **matrices de ponderación sitio específicas (Position Specific Scoring Matrices PSSMs)**

Se construyen usando algoritmos de cadenas ocultas de Markov (**HMMs**). En esencia, para un alineamiento múltiple se consideran tanto las posiciones como las frecuencias de los estados de caracter observados para cada sitio. Residuos muy conservados en una determinada posición reciben un score positivo muy alto, mientras que los raros en dicha posición reciben un score alto negativo. Resíduos que ocupan posiciones muy variables reciben scores próximos a cero.

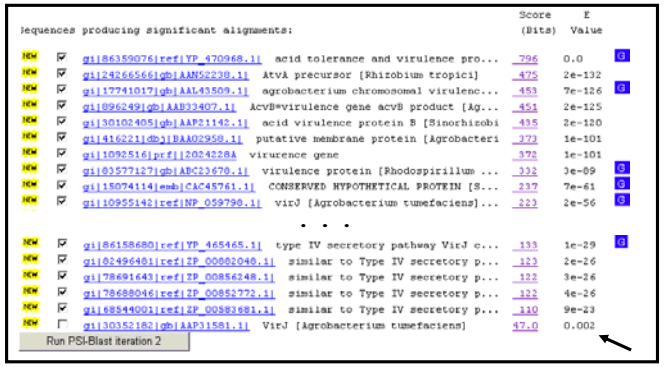


		A	R	N	D	C	Q	E	G	H	I	L	K	M	F	P	S	T	W	Y	V
1	Y	-2	-2	-2	-3	-2	-1	-2	-3	2	-1	-1	-2	-1	3	-3	-2	-2	2	7	-1
2	L	-1	-2	-3	-4	-1	-2	-3	-4	-3	2	4	-2	2	0	-3	-2	-1	-2	-1	1
3	P	-1	-2	-2	-2	-3	-2	-1	-2	-2	-3	-3	-1	-3	-4	8	-1	-1	-4	-3	-3
4	S	1	-1	0	-1	-1	0	0	-1	-1	-3	-3	0	-2	-3	-1	5	1	-3	-2	-2
5	C	-1	-4	-3	-4	9	-3	-4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

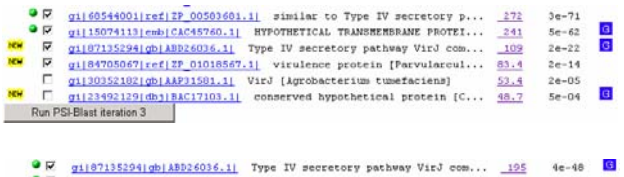
Ejemplo de una PSSM calculada para los 10 primeros resíduos de un alineamiento múltiple de proteínas HoxA de eucariontes. Sólo se muestra una pequeña parte de las secuencias incluídas en el alineamiento múltiple usado para calcular la PSSM

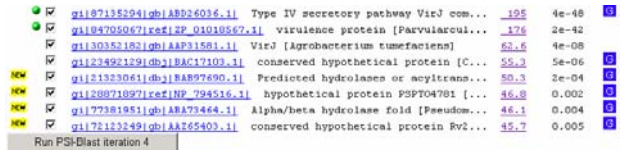
Identificación de homólogos lejanos mediante PSI-BLAST

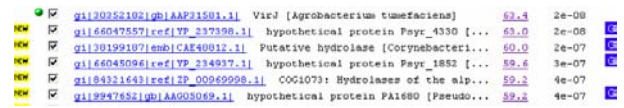
Identificación de homólogos lejanos mediante PSI-BLAST



Identificación de homólogos lejanos mediante PSI-BLAST







Identificación de homólogos lejanos mediante PSI-BLAST

- **Aspectos a cuidar al calcular PSSMs**

1.- Hay que evitar a toda costa incluir secuencias no homólogas. Revisar alineamientos pareados, estructura de dominios y no fiarse de las anotaciones. Muchas secuencias están mal anotadas !!!

Utilizar:

http://www.ncbi.nlm.nih.gov/COG/
http://psort.hgc.jp/
http://www.predictprotein.org/newwebsite/
http://www.ch.embnet.org/software/TMPRED_form.html
http://www.expasy.org/
...

para caracterizar a las proteínas dudosas ...

2.- Eliminar regiones de baja complejidad.

Usar SEG y COILS

http://www.ch.embnet.org/software/COILS_form.html

PRÁCTICAS: aprendiendo a usar PSI-BLAST para identificar homólogos lejanos

1) Descarga la secuencia Q57997 y haz un análisis de PSI-BLAST. Preguntas:
   - Qué tipo de función podría tener esta proteína?
   - Cuantos homólogos encontraste en la primera búsqueda (BLASTP)
   - Cuantos ciclos o iteraciones tuviste que correr hasta la convergencia? Cuantos homólogos pescaste?

2) Compara estos resultados con el análisis descrito en el tutorial de PSI-BLAST que encontrarás en la página del NCBI bajo: http://www.ncbi.nlm.nih.gov/Education/BLASTinfo/psi1.html

3) Ve a la página de nuestro curso y haz los ejercicios propuestos que encontrarás en el directorio Ejercicios/BLAST

Consejos finales para el uso eficiente de BLAST

0. Antes de iniciar búsquedas con BLAST, hay que **escanear las secs.** para detectar la presencia de múltiples dominios, reg. repetitivas, motivos y péptidos señal usando las herramientas o servidores apropiados (**SMART, PROSITE, PFAM, CDD, PSORT** ...)
1. Para búsquedas de secuencias homólogas distantes **usa AAs y PSI-BLAST** siempre que sea posible.
2. **PSSMs**. Usa todos los criterios adicionales que consideres relevantes para inferir la homología de manera certera. No te fíes de las anotaciones, las hay erróneas. También conviene ser crítico con las proteínas hipotéticas, puesto que su existencia no se ha demostrado experimentalmente y con frecuencia presentan extremos N terminales más largos que los de las proteínas de verdad (problema de predecir adecuadamente el inicio de traducción).
3. Ajusta el valor de los parámetros de búsqueda de manera adecuada al problema a resolver. **El valor de los parámetros determina lo que puedes encontrar**. Así por ejemplo búsquedas con NCBI-BLASTN con valores por defecto de match (+1) y mismatch (-3) tienen una frecuencia diana de 99% de identidad. No busques genes de humano y nemátodo con NCBI-BLASTN...
4. Haz **controles**, especialmente cuando se trate de similitudes en la **zona de penumbra**. Así por ejemplo puedes hacer un "**barajado" de la secuencia problema** a mano o mejor aún, usando un sencillo script de Perl. Si después de barajar los caracteres de tu secuencia sigues encontrando hits similares en la zona de penumbra, el parecido se debe simplemente a un sesgo composicional compartido entre ambas secs. y no a homología

URLs de algunas de las principales bases de datos de secuencias (DNA, Prot.), familias/dominios/motivos de proteínas y estructuras

Blocks and Blocks+ : http://blocks.fhcrc.org/
DBJ : http://www.ddbj.nig.ac.jp/
EMBL : http://www.ebi.ac.uk/embl/
Entrez : http://www.ncbi.nlm.nih.gov/Entrez/
GenBank : http://www.ncbi.nlm.nih.gov/Genbank/
InterPro : http://www.ebi.ac.uk/interpro/
MEDLINE : http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?db=PubMed
PDB : http://www.rcsb.org/pdb/
PIR : http://www-nbrf.georgetown.edu/
Pfam : http://www.sanger.ac.uk/Pfam/
PRINTS : http://www.bioinf.man.ac.uk/dbbrowser/PRINTS/PRINTS.html
ProDom : http://protein.toulouse.inra.fr/prodom.html
PROSITE :http://www.expasy.ch/prosite/prosite.html
SRS "mother" server :http://srs.ebi.ac.uk/
SWISS-PROT and TrEMBL at EBI : http://www.ebi.ac.uk/swissprot/