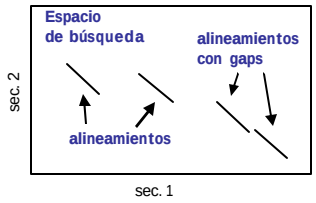
BLAST: Basic Local Alignment Search Tool

• **El algoritmo BLAST**

El **espacio de búsqueda** entre 2 secs. puedeser 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.



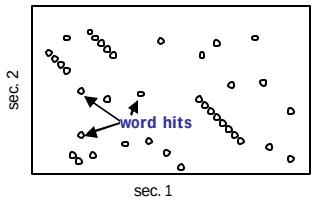
BLAST reporta todos los alns. pareados (HSPs) estadísticamente significativos encontrados en su búsqueda heurística del espacio de búsqueda. Hay que entender que en las búsquedas BLAST siempre hay que hacer un **compromiso entrevelocidad** y **sensibilidad .** La velocidad se gana al no explorar toda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

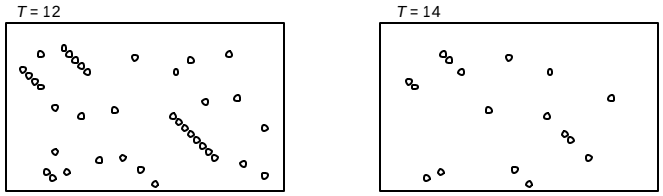
BLAST usa el concepto de **vecindad** para definir un *word hit*. Esta contiene a la palabra misma y todas las demás cuyo score sea al menos tan grande como ***T*** cuando se compara con la matriz de pesado. *T* correspondea un threshold (umbral) mínimo de score que han de tener las palabras encontradas. 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*** dependede los valores en la tabla de sustitución empleada, como del balance deseado entrevelocidad y sensibilidad. A valores más altos de *T*, menos palabras son encontradas, reduciendoel espacio de búsqueda. Ello hace las búsquedas más rápidas, a costa de incrementar el riesgo de perder algún alineamiento significativo.

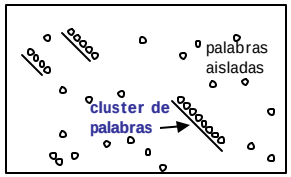


El **tamaño de palabra *W*** es otro parámetro que controla el número de word hits. *W* =1 producirá más hits que *W* = 5. Cuanto más chico sea *W* más sensible y lenta la búsqueda.
La interrelación entre *W*, *T* y la matriz de sustitución empleada es crítica, y su selección juiciosa es la mejor manera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



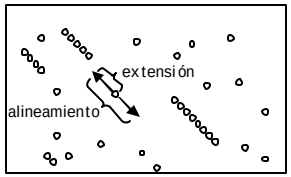
Detalles de implementación:
En **NCBI-BLASTN** las semillas son siempre palabras idénticas. *T* no es usado. Para hacer BLASTN más rápido se incrementa *W*, par hacerlo más sensible se disminuye *W*. El valor min. de *W* = 7. El algoritmo de two-hit tampoco es usado por BLASTN ya que hits de palabras 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*) entrevecinos 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 frecuentementea **regiones de baja complejidad** (rbc) que generalmente**son enmascaradas**. El uso de "**soft masking**" evita el ensemillado en rbc

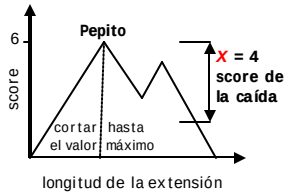
BLAST: Basic Local Alignment Search Tool

- **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)

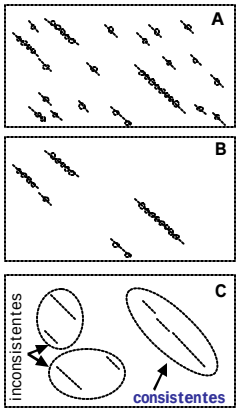
```
Pepito Pérez se fue a pescar al lago
Pepito López no vio a Arturo en casa

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**