Máster Universitario en Bioinformática

Generación y mantenimiento de datos ómicos

Curso académico 2024-25

Universidad Internacional de Valencia

Dr. Jordi Tronchoni
jordi.tronchoni@professor.universidadviu.com

06/05/2024

De:
Planeta Formación y Universidades

Tema 5
Alineamiento de secuencias

06/05/2024

Tema 1. Introducción a la bioinformática

- 1.1 Historia de la bioinformática
- 1.2 Bioética aplicada al análisis de datos

Tema 2. Principales flujos de trabajo en bioinformática

- 2.1 Genómica
- 2.2 Metagenómica y metataxonómica
- 2.3 Transcriptómica
- 2.4 Proteómica

Tema 3. Gestión de entornos y paquetes

- 3.1 Conda

Tema 4. Bases de datos y herramientas bioinformáticas

- 4.1 Principales bases de datos
- 4.2 Otros recursos online

Tema 5. Alineamiento de secuencias

- 5.1 Introducción al alineamiento de secuencias
- 5.2 Alineamientos Pairwise
- 5.3 Alineamientos Múltiples

Tema 6. Métodos de secuenciación

- 6.1 Primera generación de secuenciadores
- 6.2 Segunda generación de secuenciadores
- 6.3 Tercera generación de secuenciadores
- 6.4 Comparación de plataformas de secuenciación

Tema 7. Pre-procesado y calidad de secuencias

- 7.1 Calidad de secuencias
- 7.2 Pre-procesado de secuencias

06/05/2024

Contenidos

Tema 5. Alineamiento de secuencias

5.1 Introducción al alineamiento de secuencias

5.2 Alineamientos Pairwise

5.3 Alineamientos Múltiples

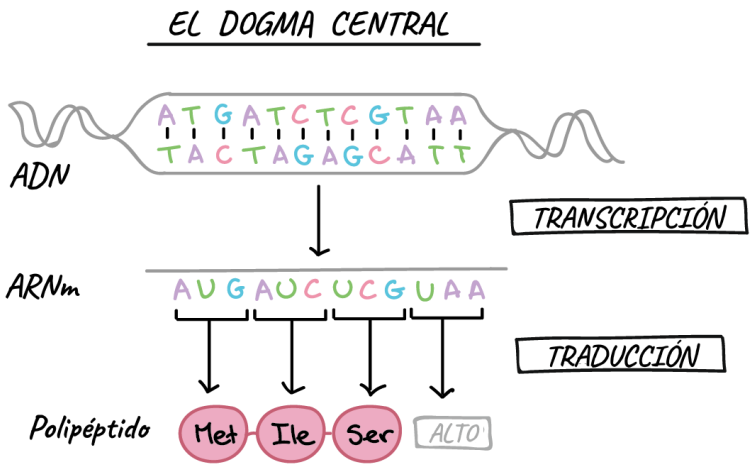
WS3_Práctica/WorkShop: Herramientas de alineamiento

06/05/2024

Recordemos

06/05/2024

Recordemos que…



06/05/2024

El dogma central de la biología molecular es un concepto fundamental que describe la transferencia de información genética en los organismos celulares. Está compuesto por tres etapas principales:

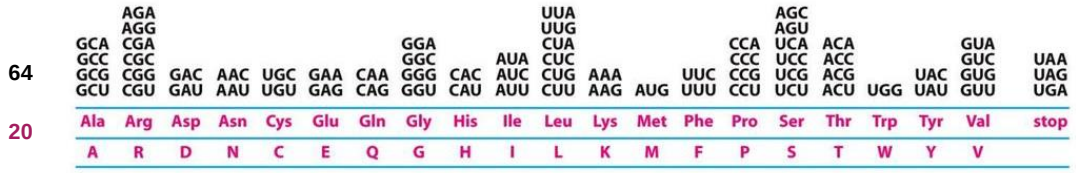
1. Replicación del ADN: En esta etapa, la doble hélice del ADN se separa y cada cadena actúa como una plantilla para la síntesis de una nueva cadena complementaria. Este proceso asegura que cada nueva célula formada tenga una copia completa y precisa del ADN.
2. Transcripción: En esta etapa, la información genética contenida en el ADN se transcribe en una molécula de ARN mensajero (ARNm). El ARNm se sintetiza utilizando una de las cadenas de ADN como plantilla, y la secuencia del ARNm es complementaria a la secuencia de ADN en la región transcrita. Esta etapa ocurre en el núcleo de las células eucariotas.
3. Traducción: En esta etapa, la información contenida en el ARNm se utiliza para sintetizar proteínas específicas. El ARNm se une a los ribosomas, que son las fábricas de proteínas de la célula, y se traduce en la secuencia de aminoácidos que formarán la proteína. Los aminoácidos son transportados al ribosoma por moléculas de ARN de transferencia (ARNt) y se unen en la secuencia específica codificada por el ARNm.

En resumen, el dogma central de la biología molecular establece que la información

genética fluye en una dirección específica: desde el ADN, que se replica, hacia el ARNm, que se transcribe, y finalmente a las proteínas, que se traducen. Este proceso es fundamental para la herencia genética y la síntesis de proteínas, que son las moléculas responsables de las funciones biológicas en los organismos.

Recordemos que…

Código genético es universal y degenerado



06/05/2024

El código genético es universal y degenerado.

El código genético es considerado **universal** porque la mayoría de los organismos utilizan el mismo código para traducir la información genética contenida en el ARN mensajero (ARNm) en secuencias de aminoácidos que formarán proteínas. Esto significa que la secuencia de nucleótidos en el ARNm determina la secuencia de aminoácidos en la proteína, independientemente del organismo en el que se encuentre.

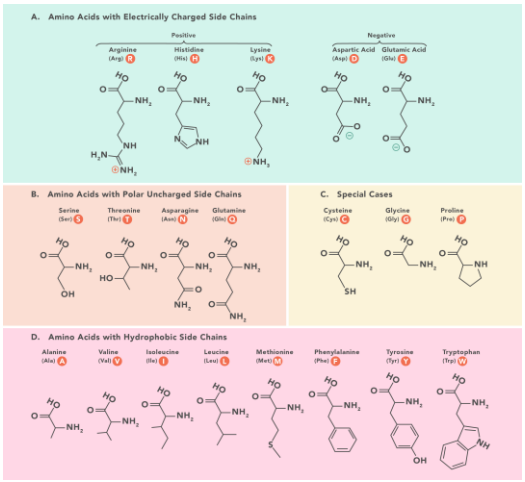
El código genético es **degenerado** porque múltiples codones (secuencias de tres nucleótidos en el ARNm) pueden codificar el mismo aminoácido. Por ejemplo, hay 61 codones diferentes que codifican los 20 aminoácidos comunes. Esto significa que algunos aminoácidos pueden ser codificados por más de un codón. Por ejemplo, el aminoácido leucina puede ser codificado por seis codones diferentes: UUA, UUG, CUU, CUC, CUA y CUG.

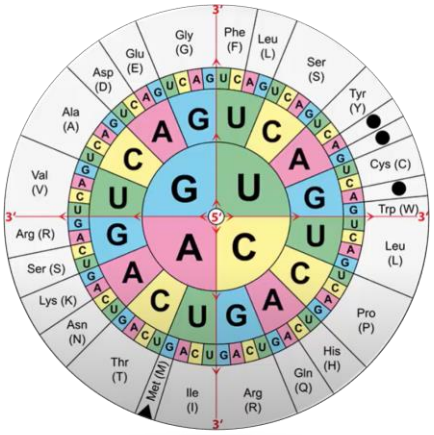
La degeneración del código genético es beneficiosa porque proporciona cierta protección contra las mutaciones. Si una mutación ocurre en un codón, es posible que el nuevo codón todavía codifique el mismo aminoácido debido a la redundancia del código. Esto reduce el impacto de las mutaciones y brinda cierta flexibilidad a la hora de cambiar las secuencias de nucleótidos sin alterar la secuencia de aminoácidos y, por lo tanto, las funciones de las proteínas.

Además, la degeneración del código genético también permite una mayor eficiencia en la síntesis de proteínas. Dado que hay múltiples codones que codifican el mismo aminoácido, la célula tiene más opciones y puede seleccionar los codones más abundantes y eficientes en función de factores como la disponibilidad de ARN de transferencia (ARNt) y la velocidad de traducción de proteínas. Solo existe un codón de inicio y codifica la metionina y existen tres de parada.
Como se observa en la figura, de los tres nucleótidos que conforman el triplete, el más variable es el tercero seguido por el segundo. Por tanto, las modificaciones o SNPs que ocurren en el tercero tienen una menor probabilidad de producir un efecto sobre la secuencia de aminoácidos.

En resumen, el código genético es considerado universal porque la mayoría de los organismos utilizan el mismo código para traducir la información genética, y es degenerado debido a que múltiples codones pueden codificar el mismo aminoácido. Esta universalidad y degeneración del código genético son características clave que permiten la diversidad y la eficiencia en la síntesis de proteínas.

Recordemos que…





06/05/2024

Algunas familias están más agrupadas que otras.
Por ejemplo los aminoácidos con carga negativa están muy agrupados, el cambio del último triplete varia entre uno y otro de la misma familia.
Los aminoácidos por su naturaleza tienen distintas cualidades y gracias a ellas se pueden agrupar en familias.
Esto añade una capa más de complejidad a la hora de valorar el efecto que puede tener un cambio de aminoácido en la cadena proteica.
Debido a estas particularidades no será lo mismo sustituir R por K que R por D.
Esperamos un mayor efecto sobre la conformación proteica con el segundo que con el primero (a priori).

01
Introducción

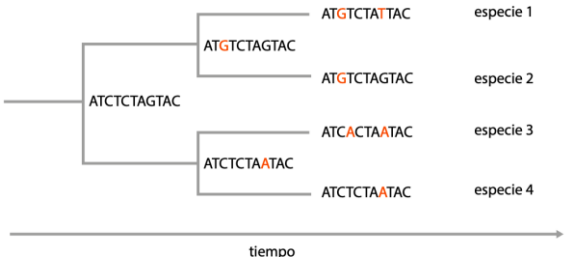
06/05/2024

¿Qué son los alineamiento de secuencias?

Los alineamientos de secuencias sirven para conocer la similitud o diferencia entre las distintas secuencias ya sea de nucleótidos o aminoácidos.

¿A que se deben estas similitudes y diferencias?

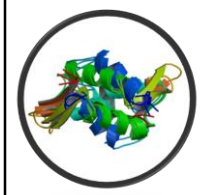
La acumulación de mutaciones en el ADN a lo largo del tiempo es la causa de que las secuencias de un mismo gen en dos especies distintas no sean idénticas. Cuanto más tiempo pase desde el último antecesor común más diferentes serán las secuencias.



06/05/2024

Los alineamientos de secuencias sirven para conocer la similitud o diferencia entre las distintas secuencias ya sea de nucleótidos o aminoácidos. Estas diferencias son debidas a la acumulación de mutaciones en el ADN a lo largo del tiempo es la causa de que las secuencias de un mismo gen en dos especies distintas no sean idénticas. Cuanto más tiempo pase desde el último antecesor común más diferentes serán las secuencias.

¿Para que sirven?



Histone H1 (residues 120-180)

```
HUMAN KKASKPKKAASKAPTKKPKATPVKKAKKKLAATPKKAKKPKTVKAKPVKASKPKKAKPVK
CHIMP KKASKPKKAASKAPTKKPKATPVKKAKKKLAATPKKAKKPKTVKAKPVKASKPKKAKPVK
MOUSE KKAAKPKKAASKAPSKKPKATPVKKAKKKPAATPKKAKKPKVVKVKPVKASKPKKAKTVK
RAT   KKAAKPKKAASKAPSKKPKATPVKKAKKKPAATPKKAKKPKIVKVKPVKASKPKKAKPVK
COW   KKAAKPKKAASKAPSKKPKATPVKKAKKKPAATPKKTKKPKTVKAKPVKASKPKKTKPVK
      ***:**********:**************  ******:****  **.**********:*  **
```

NON-CONSERVED AMINO ACIDS

Conservative | Conservative | Non-conservative | Conservative | Non-conservative | Semi-conservative | Conservative | Non-conservative

06/05/2024

Los alineamientos de secuencias en bioinformática son útiles para:

- encontrar parecidos en nuevas secuencias con secuencias de las que ya se conoce su función, identificar similitudes entre secuencias y poder cuantificarlas.
- entender las dinámicas poblacionales y las relaciones evolutivas entre genes y especies, producir árboles filogenéticos.
- Encontrar dominios funcionales mediante la identificación de regiones importantes en las secuencias.

Los alineamientos de secuencias sirven para conocer la similitud o diferencia entre las distintas secuencias ya sea de nucleótidos o aminoácidos.
Permiten conocer como dos especies han divergido en el tiempo.
Ayudan a entender que regiones son más importantes al identificar regiones con alta similitud.
Desde que somos capaces de conocer secuencias de nucleótidos o aminoácidos, su comparación entre especies o individuos han ayudado a entender su función en el organismo.
Los algoritmos de alineamiento, la forma en la que trabajan, son la base de la que ha partido la secuenciación masiva paralela y conocer como funcionan nos ayudan a entender como funciona la bioinformática más avanzada.

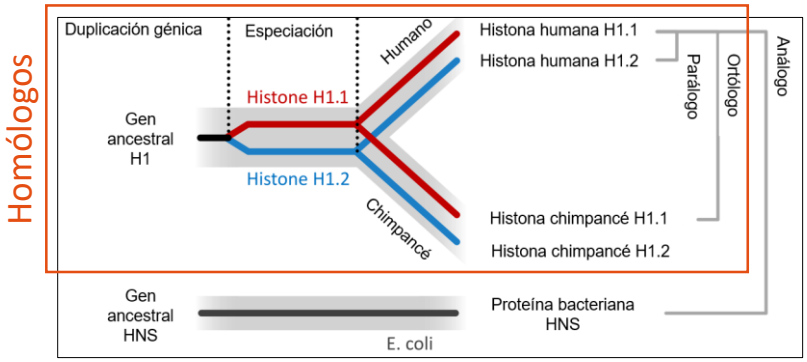
Homología

La **homología** es un concepto abstracto que deriva de la identidad de secuencia y la función proteica, prediciendo un ancestro común. Dos secuencias podrán ser o no homologas en función de si derivan de un ancestro común, independientemente de su similitud.

06/05/2024

Cuando dos secuencias son **homólogas**, significa que comparten un **ancestro común**. Generalmente cuando alineamos secuencias, estas secuencias son parecidas, tienen más puntos en común que diferencias y normalmente esto se debe a que las secuencias son homólogas. Dos proteínas son homólogas cuando provienen de un ancestro común.

Ojo, este es un término abstracto, en teoría, podríamos tener dos secuencias con alta similitud y que no fuesen homólogas (aunque no es lo habitual). A diferencia de la **similitud**, la homología no es un término cuantitativo, dos secuencias son homólogas o no lo son.



- **Parálogos**: derivan de un ancestro común pero tiene diferente función.
  Ej: Histona H1.1 y H1.2 humana.
- **Ortólogos**: derivan de un ancestro común y tienen la misma función.
  Ej: Histona H1.1 humana y de chimpancé.

- **Análogo**: no derivan de un ancestro común y realizan la misma función.
  Ej: Histona H1.1 humana y la proteína bacteriana HNS.

06/05/2024

Tomemos como ejemplo el gen de la histona H1. Los genes H1.1 y H1.2 de humanos y chimpancés son homólogos porque comparten un ancestro común.

Este gen ha ido evolucionado a lo largo del tiempo sufriendo diversos cambios. El primero es el de la especiación, dando lugar a la misma proteína en diferentes organismos, preservando la misma función. Esto sería el caso de una proteína **ortóloga**.

Por otro lado, de forma independiente se han formado las histonas 1 y 2 dentro de cada especie, dando lugar a la misma proteína pero con diferente función. Sería el caso de proteínas parálogas.

En cambio, la proteína HNS de bacteria tiene la misma función que la proteína H1 de humanos y chimapnces pero su origen en completamente distinto, con lo que sería

una proteína análoga. Comparte función pero no tienen un ancestro común.

Similitud vs Identidad

Similitud de secuencia: Considera todas las coincidencias y sustituciones.
Indica el grado de coincidencia entre dos pares de secuencias (suele expresarse en porcentajes). Incluye tanto las coincidencias exactas como las sustituciones (cuando una letra se reemplaza por otra similar).

Identidad de secuencia: Solo considera las coincidencias exactas. Indica la coincidencia total entre los pares de secuencias (similitud del 100%). La **identidad** es un **valor** que viene dado según el método de alineamiento empleado.

Una alta identidad de secuencias sugiere que el ancestro común es más reciente, mientras que una baja identidad sugiere una mayor divergencia.

Alineada

```
— C G A T G C T A G C G T A T C G T A G T C T A T C G T A C
  | | | | | | | | | | | |   | | | | | | | | | | | | |
A C G A T G C T A G C G T T T C G T A — T C — A T C G T A —
```

06/05/2024

Cuando comparamos secuencias lo hacemos con el objetivo principal de caracterizar la secuencia problema, asignarle un organismo y función. Para esto nos basaremos principalmente en el valor de **similitud** e **identidad** entre dos secuencias. Este valor se puede utilizar para inferir homología pero no es un valor directo de la misma. Si bien dos secuencias que se parecen por encima de un umbral de identidad pueden tener la misma función, no existe un valor exacto que lo indique, solo podremos decir que guardan cierta homología. El término homología se relaciona con el descendiente evolutivo, en cambio el término similitud, no necesariamente.

A pesar de lo que estamos indicando, en la práctica, cuando alinean secuencias largas, estas solo ocurren por evolución. En cambio los alineamientos cortos o pequeños si pueden deberse a la convergencia evolutiva.

Alineamiento de Secuencias

Sin alinear

```
                    CGATGCTAGCGTATCGTAGTCTATCGTAC
                                          ||
                    ACGATGCTAGCGTTTCGTATCATCGTA
```

06/05/2024

Si queremos comparar dos secuencias, lo primero que necesitamos hacer es alienarlas.
Imaginemos dos secuencias problema que queremos alinear. Las queremos alinear porque pensamos que tienen una relación. Empezamos con nucleótidos que es más sencillo.
En este ejemplo, hay alrededor de 30 nucleótidos. Enfrentadas solo coinciden en dos posiciones, indicadas por líneas verticales. Rápidamente vemos que no es un problema trivial, resulta complicado ver el alineamiento a simple vista.

Alineamiento de Secuencias

Sin alinear

```
                CGATGCTAGCGTATCGTAGTCTATCGTAC
                                ||
                ACGATGCTAGCGTTTCGTATCATCGTA
```

Alineada

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

06/05/2024

Alineamiento de Secuencias

Sin alinear

```
                CGATGCTAGCGTATCGTAGTCTATCGTAC
                                ||
                ACGATGCTAGCGTTTCGTATCATCGTA
```

Alineada

```
    C G A T G C T A G C G T A T C G T A
    | | | | | | | | | | | |   | | | | |
A   C G A T G C T A G C G T T T C G T A
```

06/05/2024

Una de las formas en las que veremos las secuencias alineadas es mediante el uso de estos guiones verticales.

Alineamiento de Secuencias

Sin alinear

```
                CGATGCTAGCGTATCGTAGTCTATCGTAC
                                ||
                ACGATGCTAGCGTTTCGTATCATCGTA
```

Alineada

```
    C G A T G C T A G C G T A T C G T A G T C T A T C G T A C
    I I I I I I I I I I I I   I I I I I I I I I I I I I I
A   C G A T G C T A G C G T T T C G T A — T C — A T C G T A —
```

06/05/2024

Este es el mejor alineamiento posible. Para conseguirlo hemos tenido que introducir “huecos” en la secuencia, los hemos representado mediante guiones horizontales.

GAPS

Sin *gaps* (10 coincidencias):

```
a:     A T A T  T G C T A C G T  A T A T C A T
          I I I  I I I I I I I  I I
b:  A T A T A  T G C T A C G T  A T C A T
```

Con *gaps* en *a* (14 coincidencias):

```
a:  A T A T  —  T G C T A C G T  A T A T C A T
    I I I I     I I I I I I I I  I I
b:  A T A T A T G C T A C G T  A T C A T
```

Con *gaps* en *a* y *b* (16 coincidencias):

```
a:  A T A T  —  T G C T A C G T  A T A T C A T
    I I I I     I I I I I I I        I I I I I I
b:  A T A T A T G C T A C G  —  —  T A T C A T
```

El objetivo del alineamiento es conseguir alinear las posiciones homólogas.

06/05/2024

Al alinear secuencias, veremos que nuestro alineamiento para el total de la secuencia mejora si introducimos espacios entre ellas. En una de las secuencias que estamos alineando o en las dos. En el ejemplo, la introducción de estos espacios o *gaps* hace aumentar las coincidencias a 16 letras. Contando las coincidencias nos es sencillo hablar de que alineamiento es el más optimo. En este caso el tercero, 16 > 14 > 10.

Puntuación de los Alineamientos

Ejemplos de sistemas de puntuación:

- Número de letras que coinciden
- Porcentaje de identidad, número de coincidencias cada cien posiciones
- Porcentaje de similitud, tiene en cuenta la similitud fisicoquímica de los diferentes aminoácidos

En la práctica se suelen utilizar sistemas de puntuación más complejos que también tienen en cuenta los *gaps*. Se suelen incluir dos penalizaciones para los *gaps*, una para abrir el *gap* y otra para extenderlo. Este último suele ser menos costoso. De entre todos los alineamientos posibles el óptimo es el que presenta una máxima puntuación para el sistema de puntuación dado.

06/05/2024

Al final se trata de un sistema de puntuación que nos permite conocer cual es el alineamiento más optimo en función de un valor. Estos sistemas de puntuación nos permiten ordenar los alineamientos de mejor a peor entre todos los alineamientos posibles.
Algunas de las variables más comunes que se tienen en cuenta en estos sistemas de puntuación son, el número de letras que coinciden (puntuación positiva al coincidir y negativa al no coincidir), el porcentaje de identidad, número de coincidencias cada 100 posiciones o el porcentaje de similitud (al alinear aminoácidos, tiene en cuenta sus características físico-químicas).

Puntuación de los Alineamientos

Ejemplo de puntuación de un alineamiento:

Sistema de puntuación: match = +1; mismatch = 0; gap = -1

```
G — A T E S L I K E S C H E E S E
I   I I I I               I I I I I I
G R A T E D — — — — — C H E E S E
```

La puntuación que le damos sería: 10 matches*1 + 1 mismatch * 0 + gaps * -1 = 4

06/05/2024

Match o coincidencia
Mismatch o falta de coincidencia
Gap o hueco

Puntuación de los Alineamientos

Ejemplo de puntuación de un alineamiento:

Sistema de puntuación: match = +1; mismatch = 0; gap = -1

```
G — A T E S L I K E S C H E E S E
I   I I I I                 I I I I I I
G R A T E D — — — — — C H E E S E
```

La puntuación que le damos sería: 10 matches*1 + 1 mismatch * 0 + gaps * -1 = 4

Pero podríamos establecer un segundo método de puntuación:

3 -1 -3 = 11

06/05/2024

La puntuación, el método de puntuación, variará en función del sistema de puntuación escogido. Distintos sistemas de puntuación serán necesarios en función del tipo de alineamiento que estamos realizando, en función de lo que comparamos (nucleótidos, aminoácidos) o el objetivo (por pares o múltiple; local o global). Distintos sistemas de puntuación o algoritmos, darán distintos valores.

Alineamientos para secuencias de aminoácidos

Dos nucleótidos pueden o ser iguales o diferentes, pero en el caso de los aminoácidos la situación es más compleja. Dos aminoácidos pueden ser iguales, diferentes o más o menos parecidos. Por ejemplo, un aminoácido neutro se parecerá más a otro aminoácido neutro que a uno ácido. Ejemplo de posible sistema de puntuación para aminoácidos:

- **Tipos de aminoácidos:**

  hidrofóbicos: Ile, Val, Leu, Ala

  Polares (+): Lys, Arg

  Polares (-): Glu, Asp

  Aromáticos: Phe, Tyr, Trp

- **Puntuaciones:**

  Ile x Val = -1

  Ile x Asp = -5

  Phe x Tyr = -1

  Phe x Gly = -8

06/05/2024

Alineamiento

- El alineamiento de secuencia a nivel de nucleótido es útil para buscar SNPs (cambios de una base) y ser más precisos en un análisis evolutivo.
- El alineamiento de secuencia a nivel de aminoácido es útil para buscar secuencias homólogas, ya que una misma secuencia proteica predice una misma función (a priori).
- Conclusión: Diferentes secuencias a nivel de nucleótido presentan una historia evolutiva pero pueden generar la misma secuencia de aminoácidos, dando la misma proteína.

06/05/2024

Existen dos tipos de alineamientos, a nivel de nucleótido y a nivel de aminoácidos. La comparación de secuencia a nivel de nucleótido es útil para estudiar filogenia y buscar diferencias singulares como la aparición de SNPs. Por otro lado, el alineamiento de secuencias de aminoácidos es útil para el estudio de homología y función de las proteínas, ya que a mayor identidad en la secuencia proteica mayor posibilidad de que se comparta la función. Son dos tipos de alineamiento que están relacionados pero que resaltan resultados diferentes.


01
Pairwise o por parejas

Pairwise

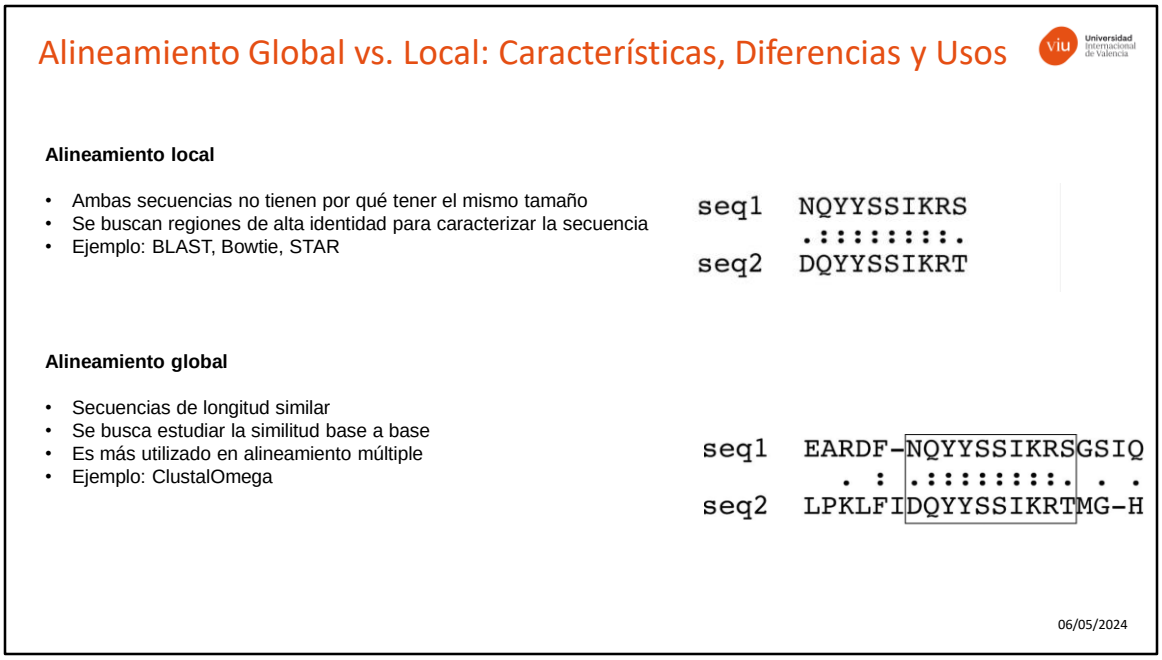
Comparación de dos secuencias mediante búsqueda de patrones de caracteres comunes.

- Alineación de dos secuencias en bases de datos
  - Caracterización de secuencias problema
- Estudios evolutivos: Las proteínas que conservan una alta **identidad** de secuencia guardan **homología**, y por tanto una función. Se pueden determinar un ancestro común.

```
AAB24882        TYHMCQFHCRYVNNHSGEKLYECNERSKAFSCPSHLQCHKRRQIGEKTHEHNQCGKAFPT 60
AAB24881        --------------------YECNQCGKAFAQHSSLKCHYRTHIGEKPYECNQCGKAFSK 40
                                    ****: .***:  * *:** * :****.:* *******..

AAB24882        PSHLQYHERTHTGEKPYECHQCGQAFKKCSLLQRHKRTHTGEKPYE-CNQCGKAFAQ- 116
AAB24881        HSHLQCHKRTHTGEKPYECNQCGKAFSQHGLLQRHKRTHTGEKPYMNVINMVKPLHNS 98
                 **** *:***********:***:**.: .****************     :  *.: :
```

06/05/2024

El alineamiento más sencillo es el alineamiento pairwise o por parejas, donde se comparan las secuencias de dos en dos. Es el alineamiento básico en la búsqueda de secuencias en bases de datos para caracterizarlas en función de la información almacenada en ellas. La comparación de secuencias puede ser a nivel de nucleótido o aminoácidos. La comparación a nivel de aminoácidos nos permite realizar estudios de **homología**, al comparar dos secuencias que sean muy similares. Estos estudios evalúan la homología de la secuencia.



Ahora bien, a la hora de realizar los alineamientos por parejas (más sencillos), existen dos enfoques principales de alineamiento: global y local. Vamos a ver sus características, diferencias y usos

Alineamiento Global:

Objetivo: Alinear toda la extensión de dos o más secuencias, maximizando la similitud a lo largo de toda su longitud.

Características:

- Busca regiones de similitud a lo largo de toda la secuencia.

- Penaliza las inserciones y deleciones (gaps) para mantener la alineación global.

- Adecuado para secuencias con alta similitud y longitud similar (Ejemplo: secuencias filogenéticamente cercanas).

Usos:

- Comparación de secuencias homólogas (con un ancestro común) para estudiar la evolución y la función.

- Identificación de mutaciones y variaciones genéticas.

- Análisis filogenético para construir árboles evolutivos.

Algoritmos: Needleman-Wunsch.

Alineamiento Local:

Objetivo: Identificar regiones de alta similitud dentro de secuencias que pueden ser de diferente longitud o tener regiones no relacionadas.

Características:

- Se enfoca en encontrar las regiones más similares, sin necesidad de alinear toda la secuencia.

- Permite gaps al inicio y final de las secuencias.

- Útil para secuencias con baja similitud global o dominios conservados.

Usos:

- Detección de motivos y dominios conservados en proteínas.

- Búsqueda de secuencias similares en bases de datos.

- Predicción de la función de proteínas basándose en la similitud con proteínas conocidas.

Algoritmos comunes: Smith-Waterman, BLAST.

El alineamiento global es ideal para comparar secuencias similares en su totalidad, mientras que el alineamiento local es mejor para identificar regiones conservadas dentro de secuencias potencialmente divergentes. La elección del método depende del objetivo del análisis y de las características de las secuencias en cuestión.

Alineamiento Global vs. Local: Características, Diferencias y Usos

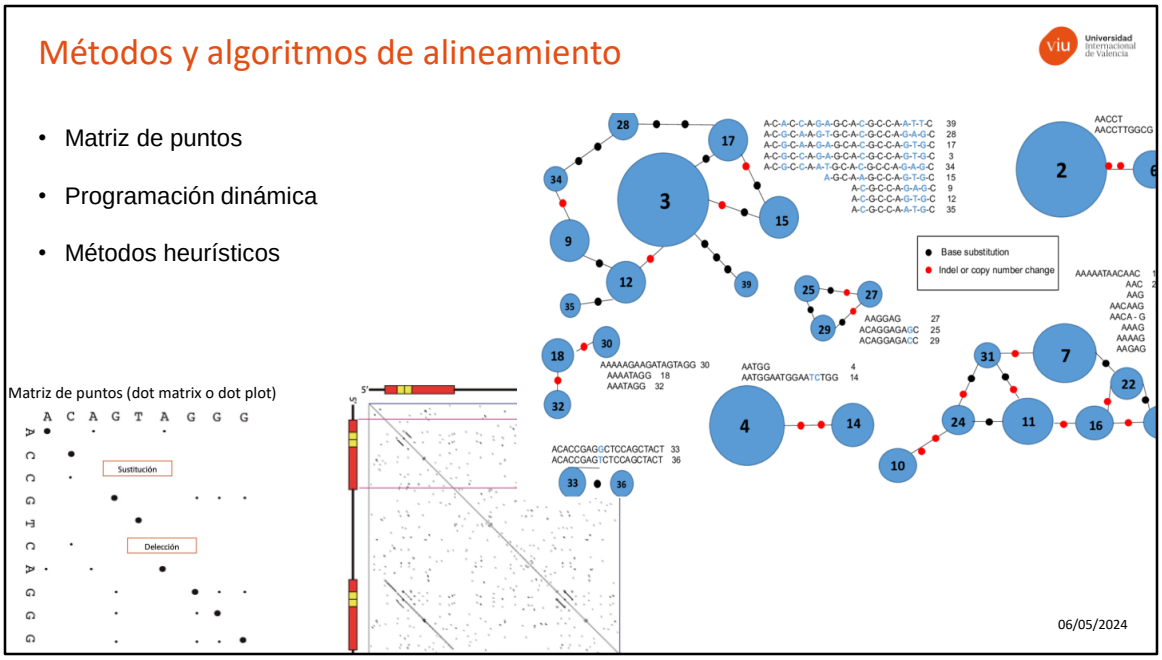
Característica	Alineamiento Global	Alineamiento Local
Objetivo	Alinear toda la secuencia	Encontrar regiones similares
Gaps	Penalizados	Permitidos, promovidos al inicio y final
Similitud	Alta similitud global	Baja o alta similitud local
Usos	Evolución, filogenia, mutaciones	Detección de motivos, búsqueda en bases de datos, predicción de función

```
Secuencia 1: ATCGGTCAGACT
Secuencia 2: ATCGGT-AGACT
```

```
Secuencia 1: MKNKFKTQEELVNQILDFLKGH
Secuencia 2: ----FKTQ--LVNQ--DFL-GH
```

06/05/2024

Matriz de puntos
Matrices de sustitución
Programación dinámica
02.1 Métodos heurísticos



Los algoritmos de alineamiento de secuencias han experimentado una evolución significativa desde sus inicios hasta la actualidad, impulsados por la necesidad de procesar datos cada vez más grandes y complejos.

Existen diferentes métodos y algoritmos para evaluar y generar alineamientos, se trata de métodos de puntuación que están basados en diferentes criterios.

Vamos a explicar cronológicamente tres métodos de alineamiento, nos ayudará a entender en que se fundamentan y porque los alineadores y mapeadores actuales funcionan de la forma en la que lo hacen.

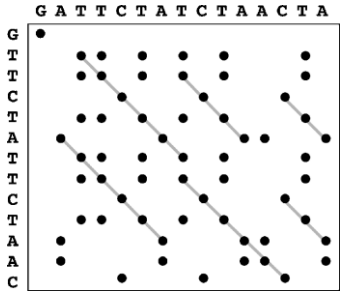
Los tres métodos que vamos a ver son: matrices de puntos, programación dinámica y métodos heurísticos. Este orden es además cronológico y nos ayudará a entender en que se fundamentan.

Matriz de puntos

- Se busca similitud base a base, colocando un punto donde se produzca.
- Muy visual, permite identificar mediante los patrones, sustituciones o repeticiones.
- Se originan diagonales en las regiones de similitud

- Útil para buscar regiones de repeticiones en una misma secuencia
- El alineamiento se visualiza en la diagonal.

Gibbs and McIntyre, 1970



06/05/2024

El método basado en matrices de puntos es el más sencillo (no necesita computación), también de los más antiguos (década de los 60). La matriz de puntos surge como una herramienta visual para identificar regiones de similitud entre secuencias biológicas de una manera intuitiva y eficiente.

Sneath y Sokal (1962): Utilizan la matriz de puntos para calcular el "coeficiente de similitud" entre secuencias. Posteriormente, en los 70, la matriz de puntos se combina con algoritmos de programación dinámica (lo veremos más adelante) para mejorar la eficiencia del alineamiento.

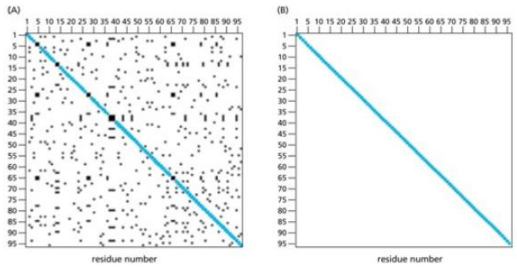
Creación de la matriz: Se crea una matriz bidimensional donde cada fila representa una secuencia y cada columna representa la posición de un carácter en las secuencias.
Marcaje de coincidencias: Se colocan puntos en las celdas de la matriz donde coinciden los caracteres de las secuencias comparadas.
Cálculo del coeficiente de similitud: Se calcula un coeficiente de similitud (SC) que

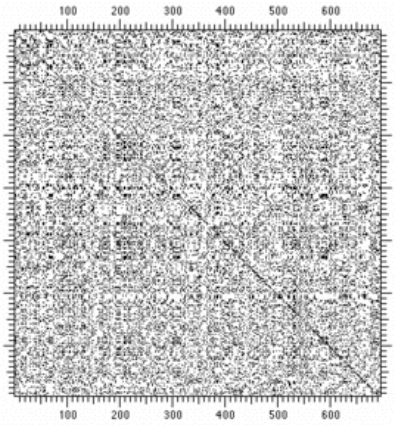
refleja la proporción de coincidencias entre las secuencias. Existen diferentes fórmulas para calcular el SC, siendo una de las más comunes: SC = (Número de coincidencias) / (Longitud máxima de las secuencias)
Interpretación del resultado: Un valor de SC alto indica una mayor similitud entre las secuencias, mientras que un valor bajo indica una menor similitud.

Recorriendo todas las filas y columnas generaremos un mapa, donde al alinear las diagonales obtendremos distintas zonas alineadas. Las diagonales de la matriz de puntos son especialmente relevantes para identificar regiones de homología entre las secuencias. Una diagonal larga y continua indica una alta probabilidad de homología. Este sistema es también útil para comparar una secuencia consigo misma y buscar regiones repetitivas. Se puede observar que el resultado es fácil de analizar visualmente, pero además podemos calcular un coeficiente de similitud con el que numéricamente comparar alineamientos.

Matriz de puntos

Ruido de fondo
Alrededor del 5% de las coincidencias son aleatorias
La solución suele ser crear una ventana por la que deslizarse de unos diez residuos.
Ejemplo de una secuencia comparada sobre si misma:





06/05/2024

Mientras el análisis sea de secuencias cortas. Al comparar dos secuencias largas vamos a tener un problema de ruido de fondo que nos va a impedir evaluar los alineamientos. Este problema se puede resolver mediante la aplicación de una "ventanas deslizantes" (nucleotide window, aminoacid window). En vez de comparar base a base, cogemos una ventana de por ejemplo 10 bases, y en función de su parecido introducimos o no el punto, es decir, comparamos regiones de 10 letras y no letra a letra.

El método de la matriz de puntos fue una herramienta fundamental en los inicios de la bioinformática, permitiendo identificar y analizar regiones de similitud entre secuencias biológicas. Aunque ha sido superado por algoritmos más sofisticados, sigue siendo un método útil para la visualización de datos de secuencias y la comprensión de las relaciones homólogas entre ellas.

Aunque se podían aplicar a distintas secuencias biológicas, ADN, ARN o proteínas, las

matrices de sustitución se introducen para ponderar las coincidencias entre aminoácidos en los alineamientos de proteínas en la década de los 70.

Matrices de sustitución

En las proteínas, ocurren muchas sustituciones que tienen poco efecto sobre la estructura o la función.
- o alteran la proteína para hacerla más adaptada a los estilos de vida de las diferentes especies
- esto depende del lugar de la proteína en el que se encuentran y de las propiedades físico – químicas de los aminoácidos.

Las **matrices de sustitución**: son puntuaciones de la probabilidad de sustituir un aminoácido por otro.

Inicialmente, las matrices de sustitución se basaban en las propiedades de los aminoácidos o en la cantidad de sustituciones de nucleótidos necesarias para cambiar de un aminoácido a otro.
Las matrices mejoraron cuando empezaron a basarse en valores reales gracias a la observación de una gran número de comparaciones.

- al estudiar dos secuencias filogenéticamente cercanas, cuanto más similares sean dos aminoácidos mayor será la probabilidad de encontrarnos con que uno **sustituya** al otro.
- observando el suficiente número de alineamientos, podremos obtener la probabilidad de que un aminoácido sea sustituido por otro basándonos en la frecuencia con la que esto ocurre en el *mundo real*.

Ahora se basan en la comparación **real** entre secuencias.
- Los dos tipos más populares: son PAM y BLOSUM
- Existen otras matrices de sustitución más especializadas, para comparar regiones transmembrana, por ejemplo.

06/05/2024

Debido a que el aumento de la dimensionalidad dificulta el análisis de los alineamientos usando los métodos anteriores, se desarrolló lo que conocemos como matrices de sustitución (o de puntuación porque damos puntos en función de la sustitución). Una matriz de puntuación reúne de antemano el valor que tiene una similitud o disimilitud. Para una cadena de ADN una similitud será un punto mientras que una diferencia será valor cero o negativo. El sistema es realmente útil cuando comparamos secuencias de aminoácidos, ya que una disimilitud no siempre es cero o negativo, hay aminoácidos que comparten propiedades y por tanto pueden tener una puntuación alta.

A lo largo de un periodo de tiempo evolutivo largo, cada aminoácido tiene más o menos probabilidad de mutar en otro aminoácido. Por ejemplo, un residuo hidrofílico como la arginina tiene más probabilidades de ser sustituido por otro residuo hidrofílico como la glutamina, que de mutar en un residuo hidrofóbico como la leucina. Esto se debe principalmente a la redundancia en el código genético, que traduce codones similares en aminoácidos similares. Además, cuando ocurre, la mutación de un aminoácido por otro con propiedades significativamente diferentes

podría afectar al plegamiento y/o a la actividad de la proteína. Siendo probable que este tipo de sustituciones se eliminen de las poblaciones por la acción de la selección natural, ya que la sustitución tiene una mayor probabilidad de hacer que una proteína no sea funcional.

Las matrices de sustitución quieren tener en cuenta en su diseño, propiedades genéticas, probabilidad de que un aminoácido sea sustituido por otro según el código genético, físico-químicas, probabilidad de que un aminoácido sea sustituido por otro según sus propiedades físico-químicas, es decir, la familia a la que pertenece, y también en evolutivas, tiene en cuenta la presión selectiva entre un cambio y otro.

Los sistemas de matrices para aminoácidos son las PAM y las BLOSUM

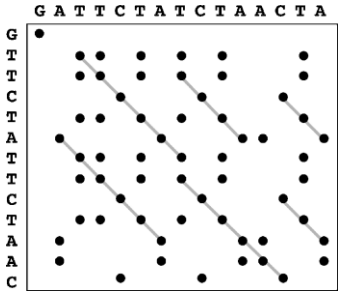
Ejemplo de matriz de sustitución:

- **Tipos de aminoácidos:**
  - hidrofóbicos: Ile, Val, Leu, Ala
  - Polares (+): Lys, Arg
  - Polares (-): Glu, Asp
  - Aromáticos: Phe, Tyr, Trp
- **Puntuaciones:**
  - Ile x Val = -1
  - Ile x Asp = -5
  - Phe x Tyr = -1
  - Phe x Gly = -8

	A	R	N	D	C	Q	E	G	H	I	L	K	M	F	P	S	T	W	Y	V	B	Z	X	*
A	4	-1	-2	-2	0	-1	-1	0	-2	-1	-1	-1	-1	-2	-1	1	0	-3	-2	0	-2	-1	0	-4
R	-1	5	0	-2	-3	1	0	-2	0	-3	-2	2	-1	-3	-2	-1	-1	-3	-2	-3	-1	0	-1	-4
N	-2	0	6	1	-3	0	0	0	1	-3	-3	0	-2	-3	-2	1	0	-4	-2	-3	3	0	-1	-4
D	-2	-2	1	6	-3	0	2	-1	-1	-3	-4	-1	-3	-3	-1	0	-1	-4	-3	-3	4	1	-1	-4
C	0	-3	-3	-3	9	-3	-4	-3	-3	-1	-1	-3	-1	-2	-3	-1	-1	-2	-2	-1	-3	-3	-2	-4
Q	-1	1	0	0	-3	5	2	-2	0	-3	-2	1	0	-3	-1	0	-1	-2	-1	-2	0	3	-1	-4
E	-1	0	0	2	-4	2	5	-2	0	-3	-3	1	-2	-3	-1	0	-1	-3	-2	-2	1	4	-1	-4
G	0	-2	0	-1	-3	-2	-2	6	-2	-4	-4	-2	-3	-3	-2	0	-2	-2	-3	-3	-1	-2	-1	-4
H	-2	0	1	-1	-3	0	0	-2	8	-3	-3	-1	-2	-1	-2	-1	-2	-2	2	-3	0	0	-1	-4
I	-1	-3	-3	-3	-1	-3	-3	-4	-3	4	2	-3	1	0	-3	-2	-1	-3	-1	3	-3	-3	-1	-4
L	-1	-2	-3	-4	-1	-2	-3	-4	-3	2	4	-2	2	0	-3	-2	-1	-2	-1	1	-4	-3	-1	-4
K	-1	2	0	-1	-3	1	1	-2	-1	-3	-2	5	-1	-3	-1	0	-1	-3	-2	-2	0	1	-1	-4
M	-1	-1	-2	-3	-1	0	-2	-3	-2	1	2	-1	5	0	-2	-1	-1	-1	-1	1	-3	-1	-1	-4
F	-2	-3	-3	-3	-2	-3	-3	-3	-1	0	0	-3	0	6	-4	-2	-2	1	3	-1	-3	-3	-1	-4
P	-1	-2	-2	-1	-3	-1	-1	-2	-2	-3	-3	-1	-2	-4	7	-1	-1	-4	-3	-2	-2	-1	-2	-4
S	1	-1	1	0	-1	0	0	0	-1	-2	-2	0	-1	-2	-1	4	1	-3	-2	-2	0	0	0	-4
T	0	-1	0	-1	-1	-1	-1	-2	-2	-1	-1	-1	-1	-2	-1	1	5	-2	-2	0	-1	-1	0	-4
W	-3	-3	-4	-4	-2	-2	-3	-2	-2	-3	-2	-3	-1	1	-4	-3	-2	11	2	-3	-4	-3	-2	-4
Y	-2	-2	-2	-3	-2	-1	-2	-3	2	-1	-1	-2	-1	3	-3	-2	-2	2	7	-1	-3	-2	-1	-4
V	0	-3	-3	-3	-1	-2	-2	-3	-3	3	1	-2	1	-1	-2	-2	0	-3	-1	4	-3	-2	-1	-4
B	-2	-1	3	4	-3	0	1	-1	0	-3	-4	0	-3	-3	-2	0	-1	-4	-3	-3	4	1	-1	-4
Z	-1	0	0	1	-3	3	4	-2	0	-3	-3	1	-1	-3	-1	0	-1	-3	-2	-2	1	4	-1	-4
X	0	-1	-1	-1	-2	-1	-1	-1	-1	-1	-1	-1	-1	-1	-2	0	0	-2	-1	-1	-1	-1	-1	-4

06/05/2024

A la derecha, ejemplo de matriz de sustitución.



No son lo mismo

	A	R	N	D	C	Q	E	G	H	I	L	K	M	F	P	S	T	W	Y	V	B	Z	X	*
A	4	-1	-2	-2	0	-1	-1	0	-2	-1	-1	-1	-1	-2	-1	1	0	-3	-2	0	-2	-1	0	-4
R	-1	5	0	-2	-3	1	0	-2	0	-3	-2	2	-1	-3	-2	-1	-1	-3	-2	-3	-1	0	-1	-4
N	-2	0	6	1	-3	0	0	0	1	-3	-3	0	-2	-3	-2	1	0	-4	-2	-3	3	0	-1	-4
D	-2	-2	1	6	-3	0	2	-1	-1	-3	-4	-1	-3	-3	-1	0	-1	-4	-3	-3	4	1	-1	-4
C	0	-3	-3	-3	9	-3	-4	-3	-3	-1	-1	-3	-1	-2	-3	-1	-1	-2	-2	-1	-3	-3	-2	-4
Q	-1	1	0	0	-3	5	2	-2	0	-3	-2	1	0	-3	-1	0	-1	-2	-1	-2	0	3	-1	-4
E	-1	0	0	2	-4	2	5	-2	0	-3	-3	1	-2	-3	-1	0	-1	-3	-2	-2	1	4	-1	-4
G	0	-2	0	-1	-3	-2	-2	6	-2	-4	-4	-2	-3	-3	-2	0	-2	-2	-3	-3	-1	-2	-1	-4
H	-2	0	1	-1	-3	0	0	-2	8	-3	-3	-1	-2	-1	-2	-1	-2	-2	2	-3	0	0	-1	-4
I	-1	-3	-3	-3	-1	-3	-3	-4	-3	4	2	-3	1	0	-3	-2	-1	-3	-1	3	-3	-3	-1	-4
L	-1	-2	-3	-4	-1	-2	-3	-4	-3	2	4	-2	2	0	-3	-2	-1	-2	-1	1	-4	-3	-1	-4
K	-1	2	0	-1	-3	1	1	-2	-1	-3	-2	5	-1	-3	-1	0	-1	-3	-2	-2	0	1	-1	-4
M	-1	-1	-2	-3	-1	0	-2	-3	-2	1	2	-1	5	0	-2	-1	-1	-1	-1	1	-3	-1	-1	-4
F	-2	-3	-3	-3	-2	-3	-3	-3	-1	0	0	-3	0	6	-4	-2	-2	1	3	-1	-3	-3	-1	-4
P	-1	-2	-2	-1	-3	-1	-1	-2	-2	-3	-3	-1	-2	-4	7	-1	-1	-4	-3	-2	-2	-1	-2	-4
S	1	-1	1	0	-1	0	0	0	-1	-2	-2	0	-1	-2	-1	4	1	-3	-2	-2	0	0	0	-4
T	0	-1	0	-1	-1	-1	-1	-2	-2	-1	-1	-1	-1	-2	-1	1	5	-2	-2	0	-1	-1	0	-4
W	-3	-3	-4	-4	-2	-2	-3	-2	-2	-3	-2	-3	-1	1	-4	-3	-2	11	2	-3	-4	-3	-2	-4
Y	-2	-2	-2	-3	-2	-1	-2	-3	2	-1	-1	-2	-1	3	-3	-2	-2	2	7	-1	-3	-2	-1	-4
V	0	-3	-3	-3	-1	-2	-2	-3	-3	3	1	-2	1	-1	-2	-2	0	-3	-1	4	-3	-2	-1	-4
B	-2	-1	3	4	-3	0	1	-1	0	-3	-4	0	-3	-3	-2	0	-1	-4	-3	-3	4	1	-1	-4
Z	-1	0	0	1	-3	3	4	-2	0	-3	-3	1	-1	-3	-1	0	-1	-3	-2	-2	1	4	-1	-4
X	0	-1	-1	-1	-2	-1	-1	-1	-1	-1	-1	-1	-1	-2	-2	0	0	-2	-1	-1	-1	-1	-1	-4

06/05/2024

A pesar de hablar siempre de matrices, debéis tener claro que una es un método de visionar los alineamientos y la otra es un método de cuantificación de las sustituciones.

Matrices PAM Dayhoff (1978)

PAM = “Point Accepted Mutations”, "Mutaciones puntuales aceptadas“:
sustituciones de un solo aminoácido que han sido "aceptadas" por la selección natural, que son funcionales en diferentes especies, basado en observaciones reales de secuencias de proteínas homólogas.

Mayoritariamente del trabajo de Dayhoff y colegas finales 1960 y principios 1970 (aunque existen algunas versiones más nuevas).

Dan una medida de la frecuencia de cambio de un aminoácido a otro, en comparación con la frecuencia de cambio aleatorio.

Derivado de alineamientos **globales** de secuencias **homólogas** de especies diferentes, pero estrechamente relacionadas (cercanas filogenéticamente). Las secuencias observadas tenían un promedio de **1 cambio de aminoácido por cada cien residuos**.

- Se realiza un análisis filogenético de las secuencias para determinar qué mutaciones se han producido
- Se calculan las puntuaciones de las tasas de cambio. Luego multiplícalos todo por 10 y redondeamos a números enteros.
- Este conjunto de puntuaciones derivadas de alineamientos de secuencias es la matriz **PAM1**.

Dado que la mayoría de las secuencias que se alinean no se encuentran entre especies tan estrechamente relacionadas, la matriz PAM1 se multiplica por sí misma muchas veces para imitar muchos cambios pequeños.

- Este concepto es una grave debilidad: la multiplicación de errores los magnifica.
- El número después de "PAM" es el número de veces que la matriz se ha multiplicado por sí misma.
- Comunes: PAM30, PAM70, PAM120, PAM250. Número más grande = mejor para relaciones más distantes

06/05/2024

Una de las primeras matrices de sustitución de aminoácidos, la matriz PAM (Point Accepted Mutation) fue desarrollada por Margaret Dayhoff en los años setenta. Esta matriz se calcula observando las diferencias en proteínas estrechamente relacionadas. Al tratarse de homólogas muy próximas, no se espera que las mutaciones observadas cambien significativamente las funciones comunes de las proteínas. Así pues, se considera que las sustituciones observadas (por mutaciones puntuales) son aceptadas por la selección natural.

Una unidad PAM se define como el 1% de las posiciones de aminoácidos que se han modificado. Para crear una matriz de sustitución PAM1, se elige un grupo de secuencias muy estrechamente relacionadas con frecuencias de mutación correspondientes a una unidad PAM. A partir de los datos mutacionales recopilados de este grupo de secuencias, se puede derivar una matriz de sustitución. Esta matriz PAM1 estima qué tasa de sustitución cabría esperar si hubiera cambiado el 1% de los aminoácidos. La matriz PAM1 se utiliza como base para calcular otras matrices asumiendo que las mutaciones repetidas seguirían el mismo patrón que las de la matriz PAM1, y que pueden producirse múltiples sustituciones en el mismo sitio. Usando esta lógica, Dayhoff derivó matrices tan altas como la PAM250. Normalmente se utilizan

las matrices PAM30 y PAM70. Una matriz para secuencias relacionadas a mayor distancia puede calcularse a partir de una matriz para secuencias estrechamente relacionadas llevando la segunda matriz a una potencia. Así se calcula la matriz PAM250. Siendo este su principal punto débil.

Principales características:

- Está basada en el código genético y propiedades fisicoquímicas de los aminoácidos

- Se basa en el alineamiento de secuencias muy relacionadas. Se observan las mutaciones entre ambas y se calcula una probabilidad. Los valores positivos son las mutaciones más frecuentes.

- Se genera la matriz PAM1. Una unidad PAM se define como el número de sustituciones si el 1% de posiciones de aminoácidos han cambiado.

- Las matrices PAMX derivan de la PAM1. Una PAM50 se origina multiplicando la probabilidad de cada aminoácido por 50.

- Al realizar un alineamiento de proteínas, la probabilidad de sustitución entre aminoácidos dependerá de la matriz PAM que usemos, en función de la similitud de las secuencias.

Matrices BLOSUM

BLOSUM = BLOck Substitution Matrix. Década de los'90 por Henikoff y Henikoff.

Basado en alineamientos **locales** de bloques, que son regiones cortas altamente homólogas entre pares de proteínas relacionadas que no pueden contener *gaps*.

Las secuencias se agruparon juntas si eran muy similares y luego se hicieron comparaciones entre los grupos como en las matrices PAM.

- Las diferentes matrices BLOSUM tienen límites específicos para las identidades de aminoácidos.
  Por ejemplo, la matriz BLOSUM62 se basa en bloques de secuencia con al menos un 62 % de identidad.
- Se calcula la probabilidad para cada sustitución, pero en lugar de tomar el logaritmo de base 10 y multiplicar el resultado por 10 como en PAM, BLOSUM toma el logaritmo de base 2 y lo multiplica por 2.

Los números más grandes implican una distancia evolutiva más cercana, por lo que BLOSUM80 es mejor para especies estrechamente relacionadas que BLOSUM45.

BLOSUM *parece funcionar mejor* que PAM siendo BLOSUM62 la matriz predeterminada que se usa en las búsquedas BLAST.

06/05/2024

La metodología de Dayhoff de comparar especies estrechamente emparentadas resultó no funcionar muy bien para alinear secuencias evolutivamente divergentes. Los cambios en las secuencias a lo largo de grandes escalas de tiempo evolutivo no se resuelven bien combinando los pequeños cambios que se producen en escalas de tiempo cortas. La serie de matrices BLOSUM (BLOck SUbstitution Matrix) corrige este problema. Henikoff & Henikoff construyeron estas matrices utilizando alineamientos múltiples de proteínas evolutivamente divergentes. Las probabilidades utilizadas en el cálculo de las matrices se calculan a partir de "bloques" de secuencias conservadas encontradas en múltiples alineamientos de proteínas. Se supone que estas secuencias conservadas tienen importancia funcional dentro de las proteínas relacionadas y, por tanto, tendrán menores tasas de sustitución que las regiones menos conservadas. Para reducir el sesgo de las secuencias estrechamente relacionadas en las tasas de sustitución, se agruparon los segmentos de un bloque con una identidad de secuencia superior a un determinado umbral, reduciendo el peso de cada uno de esos grupos (Henikoff y Henikoff). Para la matriz BLOSUM62, este umbral se fijó en el 62%. A continuación, se contaron las frecuencias de los pares entre estos grupos o clusters, por lo que sólo se contaron los pares entre segmentos

idénticos en menos de un 62%. Se utilizaría una matriz BLOSUM con un número más alto para alinear dos secuencias estrechamente relacionadas y un número más bajo para secuencias más divergentes (al contrario que en las PAM).

La matriz BLOSUM62 es muy buena detectando similitudes en secuencias distantes, y ésta es la matriz utilizada por defecto en las aplicaciones de alineamiento más recientes, como BLAST.

Algunas características:

- Criterio empírico de sustitución de aminoácidos entre proteínas relacionadas, basada en bloque de alineamientos perfectos de al menos 60 aminoácidos

- Más utilizada que PAM, mejora la fiabilidad de alineamientos locales

- En este caso se crea una matriz en cada caso. Para BLOSUM62, se crea una a partir de secuencias con una identidad del 62%. Al contrario que en PAM, cuanto más alto el valor de la matriz, más similitud existen entre las secuencias.

El sistema de matrices BLOSUM se basa en un sistema empírico de proteínas relacionadas. Las matrices están creadas a todos los niveles de identidad, con bloques de proteínas que se relacionan desde un 1% hasta el 100%, en función de su similitud. Como se puede observar el sistema de puntuación del sistema escala de forma inversa a PAM, ya que a mayor número, mayor es la identidad. La matriz BLOSUM62 es la más usada por defecto. Este sistema es muy útil para valorar alineamientos locales, ya que tiene en cuenta alineamientos en bloques de 60 aminoácidos.

Diferencias entre PAM y BLOSUM

- Las matrices PAM se basan en un modelo evolutivo explícito (es decir, las sustituciones se cuentan en las ramas de un árbol filogenético), mientras que las matrices BLOSUM se basan en un modelo implícito de evolución.
- Las matrices PAM se basan en las mutaciones observadas a lo largo de un alineamiento global, lo que incluye tanto las regiones altamente conservadas como las altamente mutables. Las matrices BLOSUM se basan únicamente en las regiones altamente conservadas en series de alineamientos que tienen prohibido contener huecos.
- El método utilizado para contar las sustituciones es diferente: a diferencia de la matriz PAM, el procedimiento BLOSUM utiliza grupos de secuencias dentro de los cuales no todas las mutaciones se cuentan igual.
- Los números más altos en el esquema de nomenclatura de la matriz PAM denotan una mayor distancia evolutiva, mientras que los números más grandes en el esquema de nomenclatura de la matriz BLOSUM denotan una mayor similitud de secuencias y, por tanto, una menor distancia evolutiva. Ejemplo: PAM150 se utiliza para secuencias más distantes que PAM100; BLOSUM62 se utiliza para secuencias más cercanas que BLOSUM50.

06/05/2024

Las matrices PAM y BLOSUM son herramientas utilizadas en bioinformática para comparar secuencias de proteínas y entender cómo han evolucionado a lo largo del tiempo.
La diferencia clave entre ellas es la forma en que consideran la evolución:
Matrices PAM (Matrices de Puntuación de Afinidad de Puntos Mutantes): Estas matrices se basan en un modelo evolutivo explícito, lo que significa que consideran las sustituciones de aminoácidos en función de un árbol filogenético que muestra cómo se cree que las especies han evolucionado con el tiempo. Es como si observaran la historia evolutiva directamente para hacer sus cálculos.
Matrices BLOSUM (Matrices de Bloques de Aminoácidos Conservados): En cambio, las matrices BLOSUM se basan en un modelo de evolución implícito, lo que significa que no consideran un árbol filogenético específico. En lugar de eso, se centran en la similitud de bloques de aminoácidos conservados en secuencias de proteínas. Estas matrices se calculan observando cuánto se conservan ciertos bloques de aminoácidos en proteínas relacionadas.

En resumen, PAM se basa en un modelo que sigue la evolución de especies específicas a lo largo del tiempo, mientras que BLOSUM se enfoca en patrones de aminoácidos conservados en proteínas relacionadas sin preocuparse por la historia

evolutiva exacta.

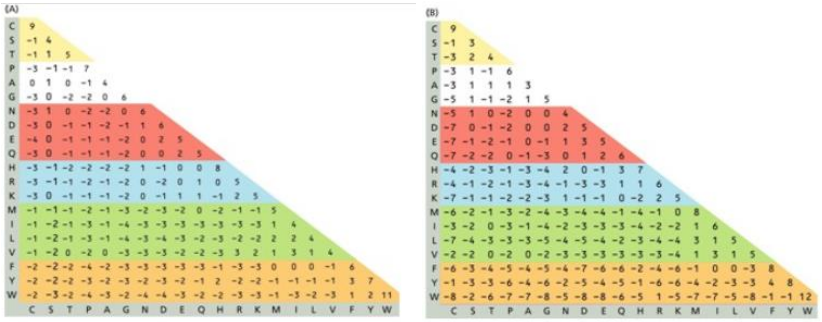
BLOSUM62 vs PAM120

Los colores representan diferentes propiedades fisicoquímicas.

Algunas sustituciones son positivas, lo que indica que ocurren con más frecuencia que el azar.

El valor medio es negativo: es más probable que un aminoácido se quede igual a que cambie.

Los valores de la diagonal son aminoácidos sin cambios, todos los cuales tienen valores positivos. Algunos son menos cambiantes que otros: especialmente el triptófano y la cisteína.



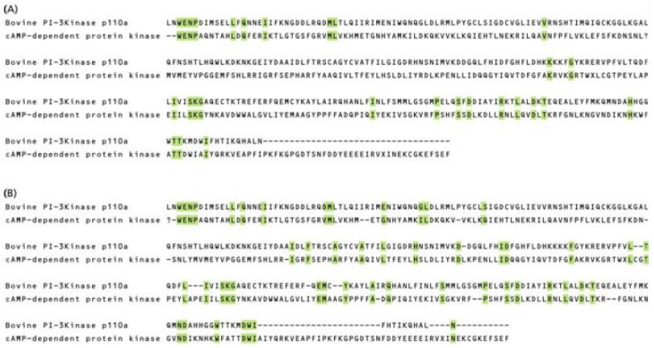
06/05/2024

Puntuaciones y penalizaciones para *gaps*

Poseen distintos sistemas de puntuación y penalización, al margen de las matrices que se utilicen.

Ejemplo:
- +1 si es una coincidencia
- 0 si no es una coincidencia
- -2 al comenzar un gap
- -1 los subsiguientes huecos

Normalmente, es más sencillo extender un gap que crearlo



06/05/2024

Los gaps o huecos representan la ausencia de caracteres en una secuencia alineada en comparación con otra. Para evaluar la calidad de un alineamiento con gaps, se utilizan diversos mecanismos de valoración, puntuación y penalización.

1. Valoración de gaps: comprender su significado biológico. Los gaps pueden indicar:
Deleciones: Pérdida de nucleótidos o aminoácidos durante la evolución.
Inserciones: Ganancia de nucleótidos o aminoácidos durante la evolución.
Intrones: Regiones no codificantes en el ADN eucariota.
Estructuras tridimensionales: Diferencias en la estructura tridimensional de las proteínas.

2. Puntuación de gaps: asigna un valor numérico a cada gap en el alineamiento. Esta puntuación refleja el "costo" de introducir un gap, considerando factores como:
Longitud del gap: Gaps más largos suelen tener una penalización mayor.
Tipo de gap: Se pueden diferenciar entre gaps de apertura (primer gap en una secuencia) y gaps de extensión (gaps adyacentes a un gap existente).
Contexto del gap: La posición del gap en el alineamiento y su proximidad a otras

características estructurales o funcionales de la secuencia pueden influir en su puntuación.

3. Penalización de gaps: mecanismo para desincentivar la introducción de gaps innecesarios en el alineamiento. Existen dos tipos de penalizaciones:
Penalización por apertura: Un valor fijo que se asigna cada vez que se introduce un nuevo gap.
Penalización por extensión: Un valor adicional que se asigna por cada carácter que se extiende un gap existente.

4. Equilibrio entre similitud y simplicidad:
La valoración, puntuación y penalización de gaps buscan un equilibrio entre dos objetivos:
Maximizar la similitud entre las secuencias alineadas: Se busca que el alineamiento refleje la mayor cantidad posible de coincidencias entre los caracteres de las secuencias.
Minimizar el número de gaps: Se busca que el alineamiento sea lo más simple posible, con la menor cantidad de huecos innecesarios.
OJO! Con el suficiente número de gaps, cualquier par de secuencias sería posible alinearla.

¿Cómo lo puntuamos?

En los sistemas de alineamiento existen diversos sistemas de puntuación a la hora de evaluar un ”gap”. Estos huecos se originan por inserciones o deleciones en las secuencias de nucleótidos que generan un desplazamiento en el alineamiento. El sistema aquí expuesto es el más básico, y se caracteriza por puntuar a los ”gaps” de forma diferente en función de si son solo un hueco o es la extensión de un gap anterior.

Los *gaps* ocurren con aproximadamente 1/10 parte de las sustituciones de bases, por lo que son comunes en la mayoría de los alineamientos.
Simbolizado por guiones (---) emparejados con residuos: como una falta de coincidencia con un espacio en blanco.

Se asigna una penalización para cada gap.
- Esto se denomina *linear gap penalty*: la penalización total es proporcional a la longitud del gap.

- El problema es que, una vez que comienzas a colocarlos, puedes alinear casi cualquier cosa (!).
- Los programas de alineamiento suelen distinguir entre crear un *gap* y ampliar un *gap* (*affine gap penalty*)

Aunque las sustituciones tienen mucha teoría detrás de ellas, las penalizaciones por *gap* generalmente se determinan por medios heurísticos (Heurística = un método o valor determinado por experimentos de prueba y error, sin una teoría guía sólida).
- En este caso, las penalizaciones por apertura y extensión de espacios son el resultado de probar muchas posibilidades y ver cuáles dan los mejores alineamientos.

El valor predeterminado de la herramienta BLAST es una penalización de -11 por abrir *gap* y -1 por cada base adicional, se resume como: *gap* (11/1).
- Otras opciones en BLAST en NCBI son 7/2, 8/2, 9/2, 10/1 y 12/1.

Una extensión común de los costes lineales estándar de los huecos es el uso de dos penalizaciones diferentes por abrir un hueco y por extenderlo. Normalmente, la primera es mucho mayor que la segunda, por ejemplo -10 por abrir un hueco y -2 por ampliarlo. Así, el número de huecos en un alineamiento suele reducirse y los residuos y huecos se mantienen juntos, lo que suele tener más sentido desde el punto de vista biológico.

Programación dinámica

¿Cómo realizamos un alineamiento?
Probar todas las posibilidades es una opción pero…son infinitas, es un problema sin fin, computacionalmente no viable.

Solución general: "**programación dinámica**", una técnica aplicada por primera vez a las secuencias de ADN por Needleman y Wunsch (1970).
- Su método original dio alineamientos globales.
- Smith and Waterman (1981) lo modifican para permitir alineamientos locales.

Estos métodos proporcionan un alineamiento óptimo para una matriz de sustitución y un conjunto de penalizaciones de gaps dados.

Aunque son mucho más rápidos que probar todas las posibilidades, continúan sin ser lo suficientemente rápidos. Varios refinamientos y métodos heurísticos mejoraran la velocidad con los años.

06/05/2024

Ya somos capaces de coger dos secuencias y alinearlas, tenemos un sistema de puntuación (matrices y *gaps*) consensuado que han demostrado ser capaces de dar valores semejantes a la realidad.

Ahora bien, comparar secuencias, en función de su longitud la cosa se complica, fácilmente llegamos a un punto donde las posibilidades son infinitas y tendríamos que calcularlas todas y comparar el resultado del sistema de puntuación para decidir.

Por tanto, ¿Cómo realizamos un alineamiento?

La técnica de **programación dinámica** puede aplicarse para producir alineamientos, globales mediante el algoritmo Needleman-Wunsch, y alineamientos locales mediante el algoritmo Smith-Waterman.

La programación dinámica es un paradigma de programación que descompone problemas complejos en subproblemas más pequeños y superponibles. Para cada

subproblema, se calcula una solución óptima y se almacena en una memoria auxiliar. Al resolver el problema original, se consultan las soluciones almacenadas de los subproblemas, evitando cálculos redundantes y optimizando el proceso. En el alineamiento de secuencias, la programación dinámica se utiliza para encontrar el alineamiento que maximiza una función de puntuación, proporcionando un alineamiento optimo. El resultado, refleja la similitud entre las secuencias, considerando la puntuación de la matriz aplicada y la penalización por gaps (huecos) en el alineamiento.

Los algoritmos de alineamiento de secuencias basados en programación dinámica, como Needleman-Wunsch y Smith-Waterman, construyen una matriz bidimensional donde cada celda representa la puntuación máxima para un alineamiento parcial entre las secuencias. Posteriormente se rellena la matriz calculando la puntuación para cada celda, considerando las puntuaciones de las celdas adyacentes y la similitud entre los caracteres de las secuencias en la posición actual. Una vez completada la matriz, se identifica el alineamiento óptimo recorriendo la matriz en sentido inverso, siguiendo el camino de puntuaciones máximas. Permite encontrar el alineamiento óptimo de manera eficiente, incluso para secuencias largas y complejas y garantiza la identificación del alineamiento con la máxima puntuación posible, pudiendo adaptarse a diferentes tipos de problemas de alineamiento, incluyendo alineamientos globales, locales y múltiples.

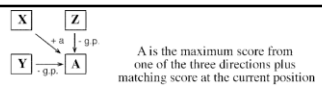
Generalmente, los alineamientos de proteínas utilizan una matriz de sustitución (PAM o BLOSUM) para asignar puntuaciones a las coincidencias o discordancias de aminoácidos y una penalización por hueco para hacer coincidir un aminoácido de una secuencia con un hueco de la otra y de esta forma rellenar la matriz de alineamiento que se ha formado en la técnica de programación dinámica . Las alineamientos de ADN y ARN pueden utilizar una matriz de puntuación más simple, pero en la práctica a menudo simplemente se asigna una puntuación positiva a las coincidencias, una puntuación negativa a los cambios y una penalización negativa a los huecos. En la programación dinámica estándar, la puntuación de cada posición de un aminoácido es independiente de la identidad de sus vecinos, por ejemplo, es posible tener en cuenta dichos efectos modificando el algoritmo, complicándolo y de esta forma teniendo en cuenta más parámetros.

De esta forma, la programación dinámica se evalúa de forma diferente en función del tipo de alineamiento que estemos llevando a cabo. Para un alineamiento local se

utiliza el algoritmo de Smith-Waterman, el cuál, es muy útil para secuencias divergente o de distinta longitud. Por su parte, el alineamiento global usa el algoritmo de Needleman-Wunsch.

Años de desarrollo de los algoritmos de alineamiento de secuencias:
Needleman-Wunsch: Desarrollado en 1968 por Saul Needleman y Christian Wunsch. Es un algoritmo de programación dinámica que busca el alineamiento global que maximiza la puntuación global, considerando tanto la similitud entre las secuencias como la penalización por gaps.
Smith-Waterman: Desarrollado en 1976 por Temple F. Smith y Michael S. Waterman. Es un algoritmo de programación dinámica que busca el alineamiento local que maximiza la puntuación local, considerando tanto la similitud entre las secuencias como la penalización por gaps.
ClustalW: Desarrollado en 1994 por Desmond Higgins y Aidan Clustal. Es un algoritmo de alineamiento múltiple que utiliza una técnica de "divide y vencerás" para mejorar la eficiencia del alineamiento de un conjunto de secuencias.

Programación Dinámica



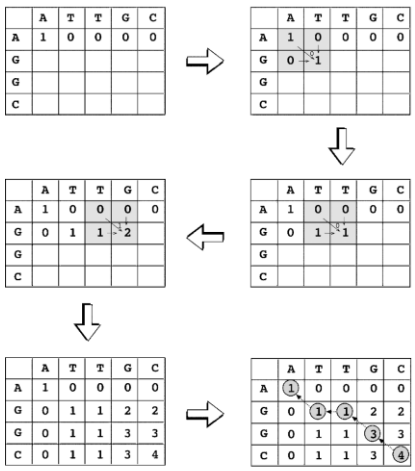
1- Se puntúa la primera fila.

2- Se puntúa la segunda fila pero teniendo en cuenta los valores previos.

3- La mejor ruta es la que presenta la mejor puntuación total, subiendo en diagonal y hacia la izquierda.

4- Si existe un desplazamiento horizontal, aparece un hueco (*gap*). Puede ser debido a una inserción o deleción.

5- Si dos caminos tienen la misma puntación, se elige uno al azar.



Final Alignment: A T T G C / A - G G C

06/05/2024

Vamos a ver un ejemplo muy simplificado. En este ejemplo se utiliza una matriz de unos y ceros para coincidencias y diferencias (si fueran aminoácidos podríamos hacerlo usando una matriz de sustitución PAM o BLOSUM). Empezando por la primera fila, colocamos un 1 en las posiciones donde existe una coincidencia y se rellena con 0 el resto. Luego la segunda fila se puntúa de la misma forma pero teniendo en cuenta las casillas que lo rodean en cada caso. Se coloca el valor máximo de una de las tres direcciones previas (N, E o NE) sumando el valor de la casilla en la que nos encontramos. Al terminar de rellenar la matriz se empieza a leer desde el valor más alto de la última fila, en diagonal y hacia la izquierda, el resultado de ese *path* o camino es nuestro mejor alineamiento. En algunos casos se producirá un desplazamiento horizontal, indicativo de un ”gap” o hueco entre las secuencias.

Un paso en diagonal es un alineamiento entre query y subject.

Un paso en vertical es un *gap* en la secuencia query

Un paso en horizontal es un *gap* en la secuencia subject.

Necesitamos un premio y una penalización por coincidencias, no coincidencias, abrir gap y extender gap.

Por otra parte, en este ejemplo hemos usado algo muy simple pero podríamos usar una matriz BLOSUM62.

Programación Dinámica

El algoritmo asigna una puntuación a cada alineamiento posible, y el propósito del algoritmo es encontrar todas las alineaciones posibles que tengan la puntuación más alta.

Needleman-Wunsch

match = 1 mismatch = -1 gap = -1

		G	C	A	T	G	C	G
	0	-1	-2	-3	-4	-5	-6	-7
G	-1	1	0	-1	-2	-3	-4	-5
A	-2	0	0	1	0	-1	-2	-3
T	-3	-1	-1	0	2	1	0	-1
T	-4	-2	-2	-1	1	1	0	-1
A	-5	-3	-3	-1	0	0	0	-1
C	-6	-4	-2	-2	-1	-1	1	0
A	-7	-5	-3	-1	-2	-2	0	0

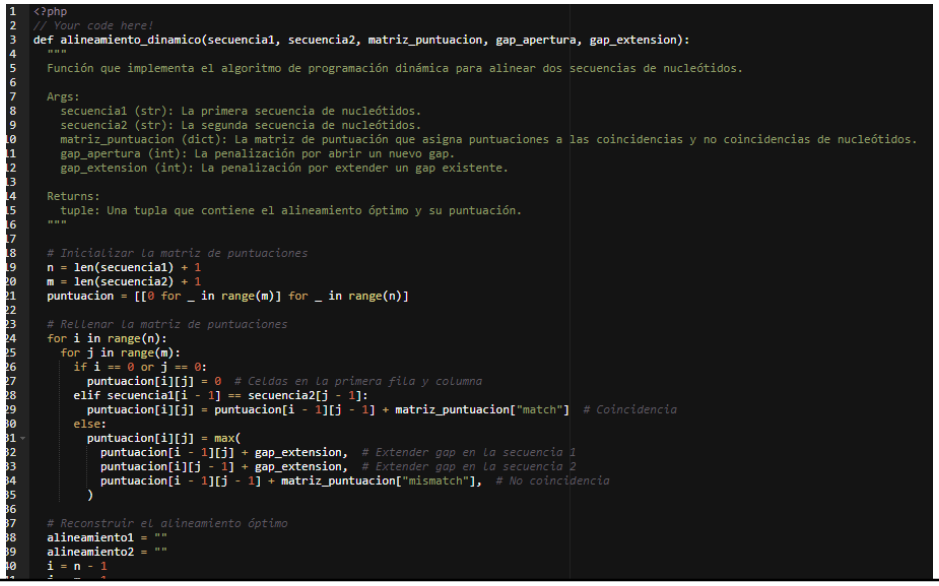
```
AlignmentA ← ""
AlignmentB ← ""
i ← length(A)
j ← length(B)
while (i > 0 or j > 0)
{
    if (i > 0 and j > 0 and F(i, j) == F(i-1, j-1) + S(A_i, B_j))
    {
        AlignmentA ← A_i + AlignmentA
        AlignmentB ← B_j + AlignmentB
        i ← i - 1
        j ← j - 1
    }
    else if (i > 0 and F(i, j) == F(i-1, j) + d)
    {
        AlignmentA ← A_i + AlignmentA
        AlignmentB ← "-" + AlignmentB
        i ← i - 1
    }
    else
    {
        AlignmentA ← "-" + AlignmentA
        AlignmentB ← B_j + AlignmentB
        j ← j - 1
    }
}
```

06/05/2024

Empezando por abajo a la derecha el algoritmo recorrerá la matriz preguntándose por las posiciones colindantes en la diagonal y escribiendo el camino en base a las puntuaciones que obtenga.

En la realidad, cuando las secuencias alcanzan la suficiente longitud, los requerimientos computacionales de memoria son desorbitados, haciendo poco útil la metodología.

Programación Dinámica



```php
<?php
// Your code here!
def alineamiento_dinamico(secuencia1, secuencia2, matriz_puntuacion, gap_apertura, gap_extension):
    """
    Función que implementa el algoritmo de programación dinámica para alinear dos secuencias de nucleótidos.

    Args:
        secuencia1 (str): La primera secuencia de nucleótidos.
        secuencia2 (str): La segunda secuencia de nucleótidos.
        matriz_puntuacion (dict): La matriz de puntuación que asigna puntuaciones a las coincidencias y no coincidencias de nucleótidos.
        gap_apertura (int): La penalización por abrir un nuevo gap.
        gap_extension (int): La penalización por extender un gap existente.

    Returns:
        tuple: Una tupla que contiene el alineamiento óptimo y su puntuación.
    """

    # Inicializar la matriz de puntuaciones
    n = len(secuencia1) + 1
    m = len(secuencia2) + 1
    puntuacion = [[0 for _ in range(m)] for _ in range(n)]

    # Rellenar la matriz de puntuaciones
    for i in range(n):
        for j in range(m):
            if i == 0 or j == 0:
                puntuacion[i][j] = 0  # Celdas en la primera fila y columna
            elif secuencia1[i - 1] == secuencia2[j - 1]:
                puntuacion[i][j] = puntuacion[i - 1][j - 1] + matriz_puntuacion["match"]  # Coincidencia
            else:
                puntuacion[i][j] = max(
                    puntuacion[i - 1][j] + gap_extension,  # Extender gap en la secuencia 1
                    puntuacion[i][j - 1] + gap_extension,  # Extender gap en la secuencia 2
                    puntuacion[i - 1][j - 1] + matriz_puntuacion["mismatch"],  # No coincidencia
                )

    # Reconstruir el alineamiento óptimo
    alineamiento1 = ""
    alineamiento2 = ""
    i = n - 1
```

06/05/2024

Ejemplo de código en Python

Problemas y mejoras de la programación dinámica

- Se pierde mucho tiempo en generar la matriz.
- La zona de alineamiento más similar puede estar muy lejos de donde empezamos a alinear.
- Seria mucho más eficiente, buscar una zona bien alineada y hacer crecer el alineamiento a izquierda y derecha de esa región.

06/05/2024

Métodos heurísticos

Herramientas para alineamientos sobre bases de datos

Se envía una secuencia (query) contra todas las secuencias de una base de datos (subjects). Se realizan los alineamientos según nuestros parámetros y se obtiene un ranking

Calcular todos los alineamientos sobre las bases de datos actuales con el método de programación dinámica (PD) llevaría mucho tiempo – Creación de métodos heurísticos, 50 – 100 veces más rápidos. Sacrifican encontrar la solución óptima a cambio de encontrar una satisfactoria y una fracción del tiempo necesario por PD.

Ejemplos:

FASTA: desarrollado por Michael Pearson en 1988
BLAST: David Lipman y Stephen Altschul en el Instituto Nacional de Salud (NIH) de los Estados Unidos a principios de la década de 1990.

06/05/2024

Una vez se desarrollan diferentes sistemas de alineamientos surgen herramientas que nos permiten comparar una secuencia problema contra una base de datos de secuencias, realizando una multitud de alineamientos. Nuestra secuencia problema se denomina ***query***, mientras que cada una de las secuencias en la base de datos se denominan ***subject***. Al ser una gran cantidad de alineamientos la capacidad de computo requerida aumenta exponencialmente, siendo inviable comparar las secuencias utilizando los métodos de matrices de punto o programación dinámica. Por esto, se desarrollan en este punto, métodos heurísticos capaces de realizar el proceso hasta 100 veces más rápido. A diferencia de los anteriores métodos, estos tienen un enfoque que se basa en reglas prácticas y estrategias aproximadas para encontrar soluciones que pueden no ser necesariamente las mejores o las óptimas, pero que son lo suficientemente buenas en un tiempo razonable. Los métodos heurísticos sacrifican la exhaustividad en favor de la eficiencia, lo que significa que pueden encontrar soluciones aceptables más rápidamente, especialmente en casos en los que la búsqueda exhaustiva de todas las posibles combinaciones es computacionalmente costosa o impracticable. Estos métodos heurísticos son particularmente útiles cuando se trabaja con secuencias largas o bases de datos grandes, donde los algoritmos de programación dinámica, que buscan soluciones óptimas, pueden ser computacionalmente prohibitivos. Los métodos heurísticos

ofrecen un equilibrio entre la calidad de la solución y el tiempo de procesamiento, lo que los hace ampliamente utilizados en la bioinformática y el análisis de secuencias biológicas. Ejemplos de estos métodos son los método FASTA y BLAST.

Herramientas para alineamientos sobre bases de datos: BLAST

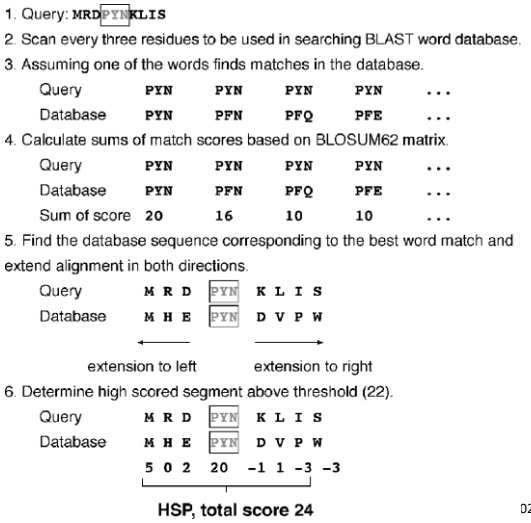
BLAST (Basic Local Alignment Search Tool)

Desarrollado en el NCBI en 1990 – La herramienta básica de alineamiento

Basado en alineamiento local de secuencias usando las matrices de sustitución→

K-mers: 3 aminoácidos, 11 nucleótidos

Los K-mers lo podemos decidir nosotros



024

BLAST (Basic Local Alignment Search Tool): BLAST es una herramienta bioinformática gratuita y de código abierto desarrollada por el National Center for Biotechnology Information (NCBI) de los Estados Unidos. Permite comparar una secuencia de consulta (query) con una gran base de datos de secuencias (subject) para identificar regiones de similitud local. Esta comparación se basa en un algoritmo heurístico que evalúa la similitud entre las secuencias mediante matrices de puntuación y modelos estadísticos.

BLAST es uno de los métodos heurísticos más ampliamente utilizados para la búsqueda y alineamiento de secuencias en bases de datos biológicas. Opera realizando búsquedas locales, lo que significa que busca regiones similares en lugar de alinear secuencias completas. BLAST utiliza una estrategia de búsqueda basada en “palabras“ (k-mers) y extensiones, lo que lo hace rápido y eficiente para buscar similitudes en grandes conjuntos de datos.

La herramienta o método BLAST nos permite comparar de una forma eficiente una secuencia respecto a una base de datos debido a que no alinea la secuencia al completo, como harían los otros métodos, si no que selecciona palabras (k-mers) que son los que se alinean (los sistema de ensamblaje funcionan de forma similar).

El proceso de búsqueda con BLAST se puede resumir en los siguientes pasos:
Preprocesamiento: La secuencia de consulta y las secuencias de la base de datos se convierten en matrices de números que representan las características de cada aminoácido o nucleótido.
Búsqueda por palabras: Se identifican coincidencias exactas o aproximadas (palabras cortas) entre la secuencia de consulta y las secuencias de la base de datos.
Extensión de alineaciones: Se extienden las coincidencias por palabras en ambas direcciones, evaluando la similitud local con una matriz de puntuación que asigna valores a las coincidencias, no coincidencias y gaps (huecos).
Filtrado: Se filtran los alineaciones por debajo de un umbral de puntuación predefinido, eliminando los menos significativos.
Evaluación estadística: Se calcula un valor E (valor esperado) que indica la probabilidad de encontrar un alineamiento con la misma o mayor puntuación por casualidad. Un valor E bajo indica un alineamiento más significativo.

En el caso de aminoácidos, el que observamos en este ejemplo, vemos cómo se realiza un alineamiento desde 3 residuos de nuestro query contra combinaciones de 3 aminoácidos de secuencias de la base de datos (k-mer= 3).

Alineamos las secuencias y puntuamos según la matriz de puntuación que hemos utilizado. Al realizar este alineamiento sobre todas las combinaciones de una secuencia si encontramos un buen alineamiento extendemos dicho alineamiento hacia los lados, hasta que la puntuación caiga por debajo de un umbral. Ese alineamiento se guardará y se realizará alineamiento respecto a otras secuencias, generando finalmente un ranking de secuencias en función de una puntuación.

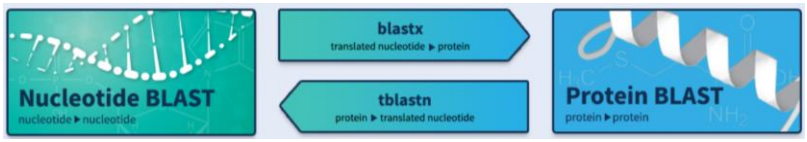
Sobre los **k-mers** o palabras K:

Un k-mer de longitud k es una subcadena de k elementos (letras o símbolos) que se extrae secuencialmente de una cadena más larga. Por ejemplo, consideremos la cadena de ADN "AGCTAGC" y establezcamos un valor de k igual a 3. Los k-mers de longitud 3 extraídos de esta cadena serían:

"AGC"
"GCT"
"CTA"
"TAG"
"AGC“

Los k-mers son útiles en el análisis genómico debido a que pueden proporcionar información sobre la estructura y las características de una secuencia. Al analizar los k-mers presentes en una secuencia de ADN o ARN, es posible identificar patrones, regiones conservadas, sitios de unión de proteínas, secuencias repetitivas, entre otros aspectos.
Además, los k-mers son utilizados en algoritmos de ensamblaje de genomas, donde se fragmenta un genoma en k-mers y se intenta reconstruir la secuencia original a partir de ellos. También se emplean en el análisis de expresión génica, la clasificación de secuencias, la búsqueda de similitudes y muchas otras aplicaciones en bioinformática.
El valor de *k* utilizado para generar los k-mers puede variar dependiendo del análisis específico que se esté realizando, el tipo de algoritmo utilizado etc.... Valores típicos de *k* oscilan entre 4 y 10, pero en algunos casos pueden ser más pequeños o más grandes, según el contexto y los objetivos de la investigación.

Herramientas para alineamientos sobre bases de datos: BLAST



Existen diferentes herramientas BLAST dependiendo de cual sea nuestro *query* y cual nuestro *subject*:

BLASTn: Nucleótido contra nucleótido (compara secuencias nucleotídicas contra una base de datos de secuencias de nucleótidos).
BLASTp: Proteína contra Proteína (compara secuencias de aminoácidos contra una base de datos de secuencias de aminoácidos).

BLASTx: Nucleótido en los seis marcos de lectura (tres marcos de lecturas por hebra) a Proteína (compara secuencias nucleotídicas contra una base de datos de aminoácidos).
tBLASTn: Proteína a nucleótido pasado a los seis marcos de lectura (secuencias de aminoácidos contra una base de datos de nucleótidos).

tBLASTx: Nucleótido a Nucleótido, pasando por todos los marcos de lectura, comparado con base de datos de nucleótidos, pasados a todas los marcos de lectura

06/05/2024

Existen diferentes herramientas BLAST dependiendo de cual sea nuestro query y cual nuestro subject.

En la plataforma del NCBI podemos encontrarlas todas y podremos realizar comparaciones de nuestras secuencias respecto a dicha base de datos.

https://blast.ncbi.nlm.nih.gov/Blast.cgi

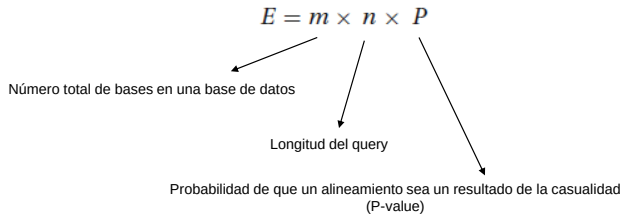
Tipos de BLAST:
BLASTP: Compara proteínas con una base de datos de proteínas.
BLASTN: Compara nucleótidos con una base de datos de nucleótidos.
TBLASTX: Traduce la secuencia de consulta de nucleótidos a proteínas y la compara con una base de datos de proteínas.
BLASTX: Compara nucleótidos con una base de datos de proteínas (sin traducción).

Además de estás, existen versiones para búsquedas muy especializadas.

Parámetros de salida de un alineamiento BLAST

1. Nombre del query
2. Nombre del subject
3. Porcentaje de identidad
4. Longitud
5. Número de mismatches
6. Gap open
7. Posición de inicio en el query
8. Posición de fin en el query
9. Posición de inicio en el subject
10. Posición de final en el subject
11. E-value
12. Bit score

E-value:
Probabilidad de que un alineamiento se de por error dependiendo del tamaño de la base de datos



Es el valor a tener en cuenta al alinear dos secuencias

06/05/2024

Al realizar un alineamiento mediante BLAST obtendremos un resultado de salida en formato de tabla con las siguientes columnas:

1. Nombre del query
2. Nombre del subject
3. Porcentaje de identidad – Grado de similitud entre dos secuencias
4. Longitud – Longitud total del alineamiento
5. Número de mismatches – Número de diferencias entre las secuencias
6. Gapopen – Número de espacios generados
7. Posición de inicio en el query
8. Posición de fin en el query
9. Posición de inicio en el subject
10. Posición de final en el subject
11. **E-value**
12. Bit score

El **e-value** es el valor más importante a tener en cuenta al analizar un alineamiento mediante Blast. El e-value calcula la probabilidad de obtener un falso positivo en nuestro alineamiento y depende de forma directa del tamaño de la base de datos utilizada. Surge de la problematica de encontrar por azar nuestra secuencia en una base de datos al aumentar el tamaño de las bases de datos de secuencias de forma exponencial en el transcurso del tiempo.

E-value vs Bit Score

Alineamiento (mapeo)

E-value: Probabilidad de que un alineamiento se de por error dependiendo del tamaño de la base de datos

A menor valor, mejor alineamiento

$$E = m \times n \times P$$

- m: Número total de bases en una base de datos
- n: Longitud del query
- P: Probabilidad de que un alineamiento sea un resultado de la casualidad (P-value)

Es el valor a tener en cuenta al alinear dos secuencias

Bit score: Valor que tiene en cuenta la identidad del alineamiento pero no es dependiente del tamaño de la base datos ni la longitud del query

A mayor valor, mejor alineamiento

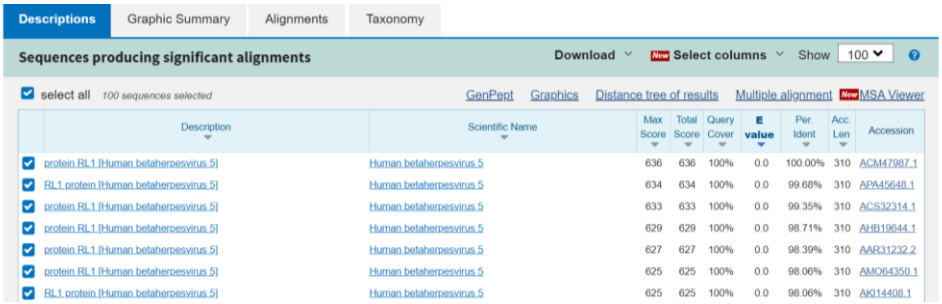
Se desarrolló para evitar el posible "problema" del ligero crecimiento de las bases de datos

06/05/2024

Por otro lado existe el valor Bitscore, muy similar al e-value. En este caso se valora la identidad y similitud de dos secuencias pero sin tener en cuenta el tamaño de la base de datos. Este valor se instauró porque se vio un posible problema en la disminución de la sensibilidad del sistema al aumentar el tamaño de la base de datos, donde dos secuencias similares dejaran de ser tan similares al aumentar la información en ellas.

BLAST

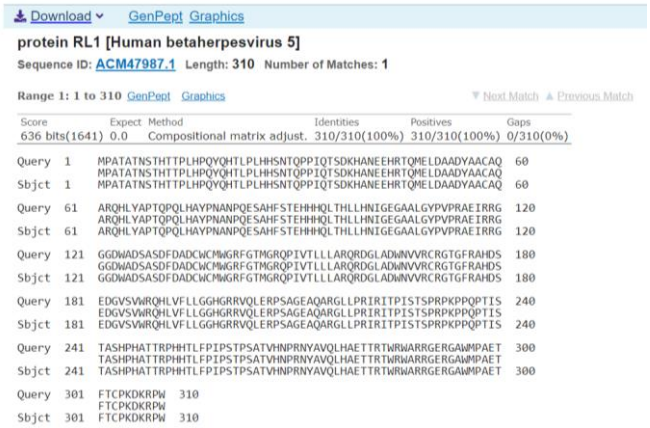
Query:
MPATATNSTHTTPLHPQYQHTLPLHHSNTQPPIQTSDKHANEEHRTQMELDAADYAACAQARQHLYAPTQPQLHAYPNANPQESAHFST
EHHHQLTHLLHNIGEGAALGYPVPRAEIRRGGGDWADSASDFDADCWCMWGRFGTMGRQPIVTLLLARQRDGLADWNVVRCRGTGF
RAHDSEDGVSVWRQHLVFLLGGHGRRVQLERPSAGEAQARGLLPRIRITPISTSPRPKPPQPTISTASHPHATTRPHHTLFPIPSTPSATV
HNPRNYAVQLHAETTRTWRWARRGERGAWMPAETFTCPKDKRPW



06/05/2024

Un ejemplo de un alineamiento mediante blast utilizando la herramienta online sería el generado por este query. Al alinear la secuencia vemos que la proteína más relacionada en la base de datos es la RL1 del betaherpesvirus humano 5. Aquí vemos que tiene un valor de e-value de 0, un porcentaje de identidad del 100% y una longitud de alineamiento de 310 aminoácidos.

BLAST



Download GenPept Graphics

protein RL1 [Human betaherpesvirus 5]

Sequence ID: ACM47987.1 Length: 310 Number of Matches: 1

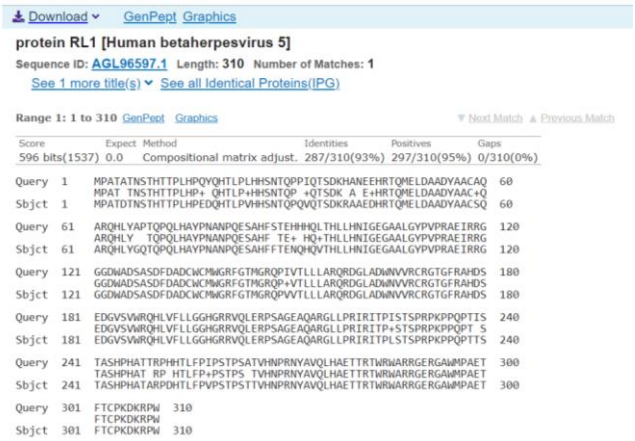
Range 1: 1 to 310 GenPept Graphics Next Match Previous Match

Score	Expect	Method	Identities	Positives	Gaps
636 bits(1641)	0.0	Compositional matrix adjust.	310/310(100%)	310/310(100%)	0/310(0%)

```
Query  1    MPATATNSTHTTPLHPQYQHTLPLHHSNTQPPIQTSDKHANEEHRTQMELDAADYAACAQ  60
            MPATATNSTHTTPLHPQYQHTLPLHHSNTQPPIQTSDKHANEEHRTQMELDAADYAACAQ
Sbjct  1    MPATATNSTHTTPLHPQYQHTLPLHHSNTQPPIQTSDKHANEEHRTQMELDAADYAACAQ  60

Query  61   ARQHLYAPTQPQLHAYPNANPQESAHFSTEHHHQLTHLLHNIGEGAALGYPVPRAEIRRG  120
            ARQHLYAPTQPQLHAYPNANPQESAHFSTEHHHQLTHLLHNIGEGAALGYPVPRAEIRRG
Sbjct  61   ARQHLYAPTQPQLHAYPNANPQESAHFSTEHHHQLTHLLHNIGEGAALGYPVPRAEIRRG  120

Query  121  GGDWADSASDFDADCWCMWGRFGTMGRQPIVTLLLARQRDGLADWNVVRCRGTGFRAHDS  180
            GGDWADSASDFDADCWCMWGRFGTMGRQPIVTLLLARQRDGLADWNVVRCRGTGFRAHDS
Sbjct  121  GGDWADSASDFDADCWCMWGRFGTMGRQPIVTLLLARQRDGLADWNVVRCRGTGFRAHDS  180

Query  181  EDGVSVWRQHLVFLLGGHGRRVQLERPSAGEAQARGLLPRIRITPISTSPRPKPPQPTIS  240
            EDGVSVWRQHLVFLLGGHGRRVQLERPSAGEAQARGLLPRIRITPISTSPRPKPPQPTIS
Sbjct  181  EDGVSVWRQHLVFLLGGHGRRVQLERPSAGEAQARGLLPRIRITPISTSPRPKPPQPTIS  240

Query  241  TASHPHATTRPHHTLFPIPSTPSATVHNPRNYAVQLHAETTRTWRWARRGERGAWMPAET  300
            TASHPHATTRPHHTLFPIPSTPSATVHNPRNYAVQLHAETTRTWRWARRGERGAWMPAET
Sbjct  241  TASHPHATTRPHHTLFPIPSTPSATVHNPRNYAVQLHAETTRTWRWARRGERGAWMPAET  300

Query  301  FTCPKDKRPW  310
            FTCPKDKRPW
Sbjct  301  FTCPKDKRPW  310
```


Al abrir el primer resultado lo primero que observamos es el alineamiento entre nuestra secuencia y la secuencia de la base de datos. Aquí vemos que es perfecto, porque es exactamente la misma proteína.

BLAST



Download GenPept Graphics

protein RL1 [Human betaherpesvirus 5]

Sequence ID: AGL96597.1 Length: 310 Number of Matches: 1

See 1 more title(s) See all Identical Proteins(IPG)

Range 1: 1 to 310 GenPept Graphics Next Match Previous Match

Score	Expect	Method	Identities	Positives	Gaps
596 bits(1537)	0.0	Compositional matrix adjust.	287/310(93%)	297/310(95%)	0/310(0%)

```
Query  1    MPATATNSTHTTPLHPQYQHTLPLHHSNTQPPIQTSDKHANEEHRTQMELDAADYAACAQ  60
            MPAT TNSTHTTPLHP+ QHTLP+HHSNTQP +QTSDK A E+HRTQMELDAADYAAC+Q
Sbjct  1    MPATDTNSTHTTPLHPEDQHTLPVHHSNTQPQVQTSDKRAAEDHRTQMELDAADYAACSQ  60

Query  61   ARQHLYAPTQPQLHAYPNANPQESAHFSTEHHHQLTHLLHNIGEGAALGYPVPRAEIRRG  120
            ARQHLY  TQPQLHAYPNANPQESAHF TE+ HQ+THLLHNIGEGAALGYPVPRAEIRRG
Sbjct  61   ARQHLYGQTQPQLHAYPNANPQESAHFFTENQHQVTHLLHNIGEGAALGYPVPRAEIRRG  120

Query  121  GGDWADSASDFDADCWCMWGRFGTMGRQPIVTLLLARQRDGLADWNVVRCRGTGFRAHDS  180
            GGDWADSASDFDADCWCMWGRFGTMGRQP+VTLLLARQRDGLADWNVVRCRGTGFRAHDS
Sbjct  121  GGDWADSASDFDADCWCMWGRFGTMGRQPVVTLLLARQRDGLADWNVVRCRGTGFRAHDS  180

Query  181  EDGVSVWRQHLVFLLGGHGRRVQLERPSAGEAQARGLLPRIRITPISTSPRPKPPQPTIS  240
            EDGVSVWRQHLVFLLGGHGRRVQLERPSAGEAQARGLLPRIRITP+STSPRPKPPQPT S
Sbjct  181  EDGVSVWRQHLVFLLGGHGRRVQLERPSAGEAQARGLLPRIRITPLSTSPRPKPPQPTTS  240

Query  241  TASHPHATTRPHHTLFPIPSTPSATVHNPRNYAVQLHAETTRTWRWARRGERGAWMPAET  300
            TASHPHAT RP HTLFP+PSTPS TVHNPRNYAVQLHAETTRTWRWARRGERGAWMPAET
Sbjct  241  TASHPHATARPDHTLFPVPSTPSTTVHNPRNYAVQLHAETTRTWRWARRGERGAWMPAET  300

Query  301  FTCPKDKRPW  310
            FTCPKDKRPW
Sbjct  301  FTCPKDKRPW  310
```

06/05/2024

Si vamos un poco más abajo en los resultados nos encontramos con resultados con menor identidad, donde se observan huecos o sustituciones. Este alineamiento, por defecto, está puntuado utilizando la matriz BLOSUM62

FASTA

- La primera herramienta desarrollada para búsquedas de similitud
- Se crean palabras cortas, de dos a seis bases y se alinean con la secuencia subject
- El sistema de alineamiento está basado en "distancias absolutas" de una cadena sobre la otra
- Mucho menos eficiente que BLAST, limitaciones como repeticiones

1. Given two amino acid sequences for comparision:

sequence 1 **AMPSDGL**
sequence 2 **GPSDNAT**

2. Construct a hashing table:

amino acid	sequence position		offset
	seq 1	seq 2	
A	1	6	-5
D	5	4	1
G	6	1	5
L	7	–	–
M	2	–	–
N	–	5	–
P	3	2	1
S	4	3	1
T	–	7	–

3. Identify residues with the same offset values (highlighted in grey).

4. Find the matching word of three residues in the order of 3, 4 and 5 in one sequence and 2, 3,and 4 in the other.

5. This allows establishment of alignment between the two sequences.

sequence 1 **AMPSDGL-**
sequence 2 **-GPSDNAT**

06/05/2024

FASTA: Similar a BLAST, el algoritmo FASTA también es una herramienta heurística que se utiliza para buscar secuencias similares en bases de datos biológicas. Utiliza un enfoque de búsqueda local y utiliza estrategias de palabras clave y extensiones para encontrar alineamientos.

Esta herramienta es anterior a BLAST y está en desuso por ser menos potente. Este método realiza alineamientos completos y calcula la puntuación en función de las posiciones desplazadas entre cada uno de los aminoácidos. Cogiendo los valores de distancia más bajos (offset) se puede realizar un alineamiento.

La principal diferencia entre BLAST y FASTA es que BLAST se dedica sobre todo a encontrar alineamientos de secuencias óptimas a nivel local sin *gaps* (o lo menos posible), mientras que FASTA se dedica a encontrar similitudes entre secuencias menos parecidas.

03
Alineamientos múltiples

Alineamiento múltiple

- Alineamiento de más de dos secuencias que guardan una relación evolutiva
- Caracterización de regiones conservadas
- Búsqueda de homólogos
- Alineamiento básico para análisis de filogenia
- Al igual que pasaba con los alineamientos BLAST y FASTA, solo podría seguirse un método basado en programación dinámica si el número de secuencias es bajo
- Se sigue utilizando las matrices de sustitución (BLOSUM62)
- Tres tipos de alineamientos posibles: **progresivo**, **iterativo** y **basado en bloques**

06/05/2024

Además de los sistemas de alineamiento por parejas existen los alineamientos múltiples, donde un bloque de secuencias se alinean simultáneamente.

Es útil para alineamiento de secuencias de nucleótidos y aminoácidos, siendo especialmente relevante este último para estudios de filogenia.

La principal utilidad de estos alineamientos es el estudio evolutivo de las secuencias, a través del alineamiento de secuencias homólogas y conservadas entre distintas especies localizando los dominios importantes de función en proteínas.

En este método se siguen utilizando las matrices de puntuación mediante el sistema de ”suma por pares”, que calcula el valor entre todas las combinaciones de bases alineadas.

Existen tres tipos de alineamientos múltiples posibles: progresivo, iterativo y basado en bloques.

Objetivos

Las aplicaciones más habituales de los alineamientos múltiples son:

- la reconstrucción filogenética,
- el análisis estructural de proteínas,
- la búsqueda de dominios conservados y
- la búsqueda de regiones conservadas en promotores.

06/05/2024

Algunas de las aplicaciones más comunes incluyen:

1. Filogenia Molecular: Los alineamientos múltiples se utilizan para construir árboles filogenéticos que representan las relaciones evolutivas entre diferentes especies. Al comparar las diferencias y similitudes en las secuencias genéticas, se puede inferir la historia evolutiva y la diversidad de las especies.

2. Análisis de Homología: Los alineamientos múltiples permiten identificar genes homólogos, es decir, genes relacionados por descendencia evolutiva común. Esto es esencial para la predicción de funciones de genes desconocidos y la comprensión de las similitudes y diferencias genéticas entre especies.

3. Análisis de Regiones Conservadas: Los alineamientos múltiples ayudan a identificar regiones genómicas o proteicas conservadas en diferentes especies. Estas regiones conservadas suelen estar relacionadas con funciones biológicas importantes y se utilizan en la anotación de genes y la identificación de elementos reguladores.

4. Identificación de Motivos y Dominios: Los alineamientos múltiples se utilizan para identificar motivos o dominios conservados en secuencias proteicas. Esto es útil para

predecir funciones de proteínas, como la unión a ligandos o la interacción con otras proteínas.

5. Análisis de Evolución Molecular: Al comparar alineamientos múltiples de secuencias de genes o proteínas, es posible estudiar cómo han evolucionado a lo largo del tiempo. Esto proporciona información sobre los eventos de duplicación génica, mutaciones y adaptaciones a diferentes entornos.

6. Análisis de Variabilidad Genética: Los alineamientos múltiples se utilizan para identificar polimorfismos de nucleótidos únicos (SNPs) y otros tipos de variabilidad genética en poblaciones. Esto es esencial en estudios de genética de poblaciones y asociación de variantes con enfermedades.

7. Diseño de Cebadores y Sondas: Los alineamientos múltiples se utilizan para diseñar cebadores y sondas específicos para la amplificación o detección de secuencias genéticas de interés en técnicas como la PCR o la hibridación in situ.

8. Análisis de Expresión Génica: Al comparar secuencias de genes y sus regiones reguladoras en múltiples especies, es posible estudiar la expresión génica y las diferencias en la regulación génica entre diferentes organismos.

Estas son solo algunas de las aplicaciones más comunes de los alineamientos múltiples en bioinformática genómica. Estos análisis son fundamentales para comprender la estructura y función de genomas y proteomas, así como para abordar preguntas de evolución, variabilidad genética y biología comparativa.

Alineamiento Progresivo

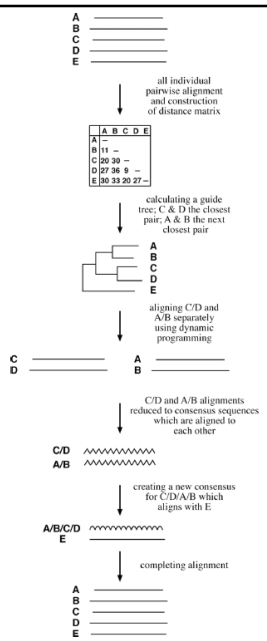
1- Se realiza un alineamiento global por pares siguiendo la estrategia Needleman–Wunsch

2- Se crea una matriz de distancia basándonos en la similitud de las secuencias

3- Las dos secuencias más similares son las guías. Se crea una **secuencia consenso** entre ellas y se alinea otra secuencia sobre esa nueva secuencia consenso

Posibles problemas: Las dos primeras secuencias generen un mal alineamiento

Ejemplo: ClustalW, MAFFT, MUSCLE



06/05/2024

Alineamiento Múltiple Progresivo:
Enfoque: El método progresivo construye el alineamiento paso a paso, comenzando con la alineación de dos secuencias y luego agregando secuencias adicionales una a una. Se basa en la jerarquía de alineamientos.
Estrategia: En cada paso, las secuencias ya alineadas se tratan como una sola secuencia "consenso" y se alinean con la siguiente secuencia de entrada. Este proceso se repite hasta que se alinean todas las secuencias.
Ejemplo: Un ejemplo de alineamiento progresivo es el método ClustalW.

Pasos:
Selección de pares de secuencias: Se seleccionan dos secuencias del conjunto y se alinean utilizando un método de alineamiento de pares, como la programación dinámica o heurísticas.
Construcción de un árbol guía: Se construye un árbol filogenético que representa las relaciones de similitud entre las secuencias. Este árbol guía sirve como referencia para el alineamiento posterior.
Alineación progresiva: Se alinean secuencias adicionales al alineamiento inicial en función del árbol guía. Se inserta cada nueva secuencia en la posición más probable, considerando su similitud con las secuencias ya alineadas.

Refinamiento del alineamiento: Se realizan ajustes al alineamiento para optimizar la puntuación global y minimizar los gaps (huecos).

Iterativo

Se genera una solución óptima a partir de soluciones rápidas de baja calidad y evaluándolas

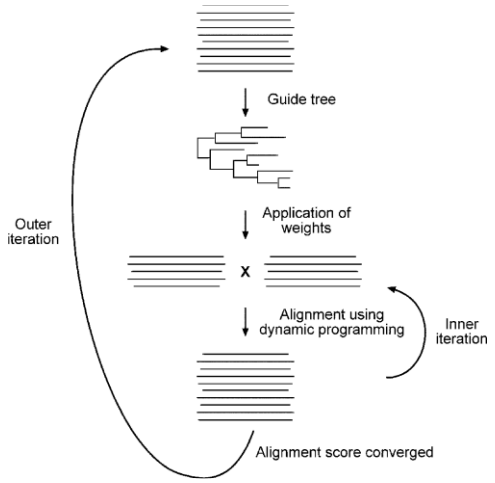
1- Se realizan alineamientos entre todas las secuencias sin criterios inicial

2- Se evalúan dichos alineamientos, con herramientas como machine learning o búsqueda de motivos con *Hidden Markov Models**

3- Se vuelve al inicio guardando los *gaps* generados y se vuelve a alinear hasta conseguir una solución estable

Posibles problemas: Bloqueo en un mínimo y no llegar a la solución

Ejemplo: Clustal Omega



06/05/2024

Alineamiento Múltiple Iterativo:
A diferencia del Alineamiento Múltiple Progresivo, que construye el alineamiento de forma gradual a partir de pares de secuencias, el AMI utiliza un enfoque **iterativo** que refina un alineamiento inicial de manera global.

Enfoque: El método iterativo comienza con un alineamiento inicial, que puede ser generado por un método progresivo o por un método heurístico. Luego, se mejora iterativamente el alineamiento, reajustando las secuencias para optimizar la puntuación del alineamiento.
Estrategia: Después del primer alineamiento, las secuencias se reeligen en función de sus similitudes y se vuelven a alinear en cada iteración. Este proceso se repite hasta que la calidad del alineamiento converge.
Ejemplo: El algoritmo MAFFT es un ejemplo de un método de alineamiento múltiple iterativo.

Mediante este método se generan inicialmente alineamientos entre secuencias aleatorias del sistema y se valora el bloque final resultado. Se realiza el sistema de

nuevo y se va guardando la información sobre los mejores alineamientos que se generaron en el paso anterior. Las interacciones que sufre el sistema le permite aprender cual son los mejores alineamientos. Este sistema se repetirá hasta llegar a un mínimo, una solución estable, existiendo el potencial problema de que nunca se llegue al mismo. Uno de los ejemplos más relevantes es el de ClustalOmega, una evolución de ClustalW.

Pasos:

Obtención de un alineamiento inicial: Se utiliza un método de alineamiento de pares o un alineamiento aleatorio como punto de partida.

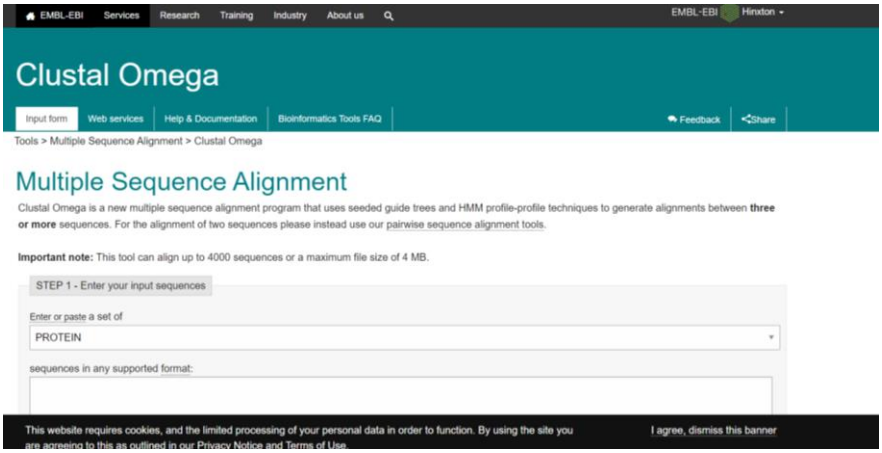
Evaluación del alineamiento: Se calcula una puntuación que refleja la calidad del alineamiento, como la suma de los gaps o la divergencia entre las secuencias.

Optimización del alineamiento: Se realizan cambios locales en el alineamiento para mejorar la puntuación, como mover segmentos de secuencias o insertar gaps.

Repetición de los pasos 2 y 3: Se repiten la evaluación y optimización del alineamiento hasta que no se observen mejoras significativas en la puntuación.

*La búsqueda de motivos con HMM es una técnica que utiliza modelos estadísticos para identificar patrones o motivos conservados en secuencias biológicas.

Iterativo

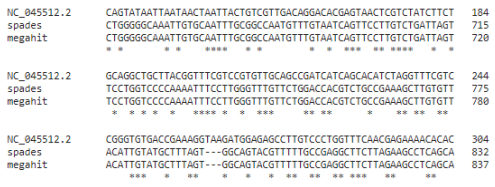


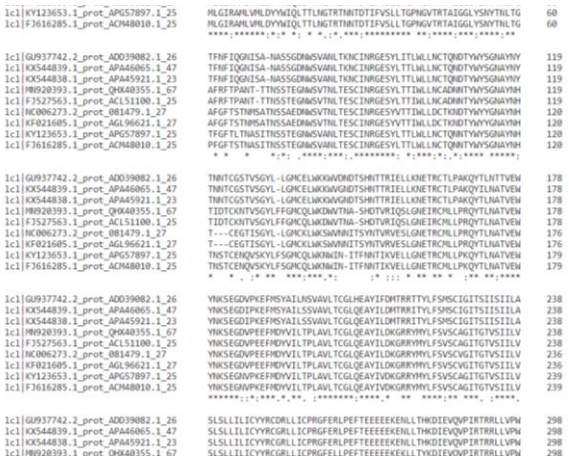
06/05/2024

Aquí podemos ver la herramienta online de ClustalOmega, dentro de la plataforma del EBI. Aquí podemos alinear secuencias de nucleótidos y proteínas.

https://www.ebi.ac.uk/Tools/msa/clustalo/

Clustal Omega





06/05/2024

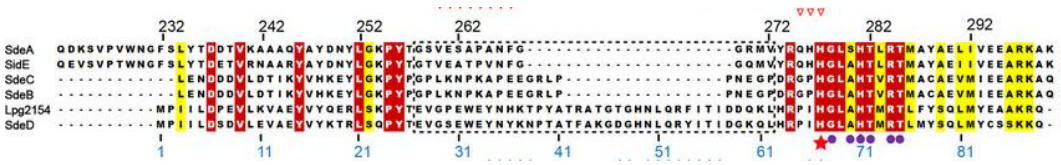
Este sería el resultado. A la izquierda vemos un bloque de alineamiento de nucleótidos y la derecha un bloque donde hemos alineado aminoácidos.

Basado en Bloques

Único método basado en alineamiento local.

Se identifica un bloque conservado entre las secuencias (de longitud variable) y se alinea siguiendo el sistema de suma por pares.

Poco usado, sin apenas utilidad.



06/05/2024

Alineamiento Múltiple Basado en Bloques:
El último criterio de alineamiento se denomina basado en bloques. En este sistema se originan alineamientos locales entre las secuencias, buscando bloques conservados y evaluándolos según el sistema de matrices de puntos y suma por pares. Este sistema apenas se usa porque realmente está basado en alineamientos locales. A diferencia del alineamiento múltiple progresivo (AMP) y el alineamiento múltiple iterativo (AMI), que alinean las secuencias en su totalidad, el AMB se enfoca en **identificar y alinear regiones conservadas** entre las secuencias, también llamadas bloques.
Enfoque: Este enfoque divide las secuencias de entrada en bloques antes de realizar el alineamiento. Cada bloque contiene secuencias que se cree que tienen una relación evolutiva cercana.
Estrategia: Cada bloque se alinea por separado, y luego los bloques alineados se ensamblan en un alineamiento múltiple completo. Esta estrategia se utiliza para lidiar con secuencias altamente divergentes o con estructura de dominio compleja.
Ejemplo: El programa GBlocks es un ejemplo de un método de alineamiento múltiple basado en bloques.

Pasos:

Detección de bloques: Se identifican regiones similares (bloques) en las secuencias utilizando algoritmos de búsqueda por palabras o métodos estadísticos.

Alineamiento de bloques: Se alinean los bloques identificados, considerando su similitud y posición relativa en las secuencias.

Inserción de gaps: Se insertan gaps (huecos) en las secuencias para optimizar el alineamiento de los bloques y minimizar los gaps entre ellos.

Refinamiento del alineamiento: Se realizan ajustes locales en el alineamiento para mejorar la puntuación global y minimizar los gaps.

Diferencias clave

Enfoque de construcción: En el progresivo, se construye el alineamiento de manera incremental. En el iterativo, se parte de un alineamiento inicial y se refina iterativamente. En el basado en bloques, se dividen las secuencias en bloques y se alinean por separado antes de ensamblar el alineamiento final.

Estrategia: El progresivo se basa en la jerarquía de alineamientos, el iterativo busca la convergencia y mejora de la calidad, y el basado en bloques se centra en segmentos conservados.

Aplicación: El enfoque progresivo es útil para secuencias estrechamente relacionadas. El iterativo es efectivo para refinar alineamientos iniciales. El basado en bloques es útil cuando se tienen secuencias altamente divergentes o con estructura de dominio compleja.

06/05/2024

Enfoque de construcción: En el progresivo, se construye el alineamiento de manera incremental. En el iterativo, se parte de un alineamiento inicial y se refina iterativamente. En el basado en bloques, se dividen las secuencias en bloques y se alinean por separado antes de ensamblar el alineamiento final.

Estrategia: El progresivo se basa en la jerarquía de alineamientos, el iterativo busca la convergencia y mejora de la calidad, y el basado en bloques se centra en segmentos conservados.

Aplicación: El enfoque progresivo es útil para secuencias estrechamente relacionadas. El iterativo es efectivo para refinar alineamientos iniciales. El basado en bloques es útil cuando se tienen secuencias altamente divergentes o con estructura de dominio compleja.

La elección entre estos enfoques depende de la naturaleza de las secuencias y el objetivo del análisis de alineamiento múltiple. Cada uno tiene sus propias ventajas y desventajas.

Problemas de los alineamientos por pares o múltiples

- Los alineamientos deberían plasmar la línea evolutiva: las mutaciones reales que han ocurrido y que han llevado a la selección de ese gen.
  - Pero en realidad es demasiado complicado conseguirlo
  - Los alineamientos de proteínas funcionan mejor sobre mayores distancias evolutivas que los de nucleótidos.
- Complicado tomar decisiones sobre las puntuaciones en sustituciones, inserciones y deleciones (gaps).
- Complicado tomar decisiones sobre los posibles scores de los alineamientos globales y locales.
- Estrategias de alineamiento múltiple alejadas de alineamiento por pares.

Recordemos que, el objetivo de un alineamiento de secuencias es hacer coincidir "**los elementos más similares**" de dos secuencias.

06/05/2024

Al final seremos nosotros los que tendremos que revisar ese alineamiento y verificar como de satisfactorio es.


WS4
Alineamientos

WS4_Alineamientos

#WS4 Alineamientos de secuencias
#1 Alineamiento óptimo para las siguientes secuencias
#2 Descargar una secuencia
#3 BLAST WEB
#4 BLAST local
#5 Alineamiento múltiple de secuencias mediante ClustalOmega
#en web
#en local
#5b Alineamiento múltiple de secuencias mediante MUSCLE
#6 Dominios importantes en proteínas, caso LACTASE

06/05/2024

¡Gracias!


viu
Universidad
Internacional
de Valencia
universidadviu.com
De:
Planeta Formación y Universidades