Universidad de Las Palmas de Gran Canaria

Facultad de Informática

Práctica 4: Búsqueda de Similitud de Secuencias con BLAST y Biopython

Bioinformática – Curso Académico 2025–2026

Nombre del Grupo: BioTeam 2025

Autores:

Raúl Mendoza Peña (DNI: XXXX5580F)
Adrián Ojeda Viera (DNI: XXXX4780V)

Repositorio:

https://github.com/raulmendoza21/BIOINFORMATICA-BLAST.git

Fecha de Entrega: 5 de diciembre de 2025

"Este informe detalla la automatización de búsquedas BLAST (Basic Local Alignment Search Tool) utilizando la librería Biopython. Se comparan ejecuciones remotas (servidores NCBI) contra ejecuciones locales mediante BLAST+ y subprocess, analizando métricas de identidad, cobertura y significancia estadística (E-value)."

Resumen

En esta práctica de laboratorio se ha abordado el uso de la herramienta BLAST para la búsqueda de alineamientos locales de secuencias biológicas. Se desarrollaron scripts en Python utilizando el módulo `Bio.Blast` de Biopython para interactuar con la API pública del NCBI (búsqueda online) y se implementó una solución robusta utilizando `subprocess` para ejecutar la suite BLAST+ instalada localmente (búsqueda local). Se generaron bases de datos propias de nucleótidos y proteínas mediante `makeblastdb`. Los resultados obtenidos permitieron comparar la eficiencia y flexibilidad de ambos entornos, aplicando filtros avanzados por E-value y taxonomía, y validando la corrección biológica mediante el uso de secuencias reales (Insulina y Hemoglobina) frente a secuencias aleatorias.

1. Introducción

BLAST (*Basic Local Alignment Search Tool*) es el algoritmo fundamental en bioinformática para comparar información biológica primaria. Permite encontrar regiones de similitud local entre secuencias, lo que es crucial para inferir relaciones funcionales y evolutivas.

Los objetivos principales de esta sesión han sido:

1. **Automatización:** Utilizar `Biopython` para enviar consultas a los servidores del NCBI sin intervención manual en la interfaz web.
2. **Entorno Local:** Configurar y ejecutar BLAST+ localmente, gestionando bases de datos personalizadas para nucleótidos y proteínas.
3. **Análisis Crítico:** Interpretar métricas clave como el *E-value* (probabilidad de alineamiento por azar), el porcentaje de identidad y la cobertura del alineamiento (*Query Coverage*).

2. Ejercicio 1: BLASTN (ADN)

El objetivo fue realizar una búsqueda de nucleótidos (`blastn`) tanto online como localmente a partir de una secuencia introducida por teclado.

2.1. Implementación Online

Se utilizó la función `NCBIWWW.qblast` apuntando a la base de datos estándar `nt`. Se implementó un manejo de excepciones para tolerar fallos de red.

```
result_handle = NCBIWWW.qblast(
    program="blastn",
    database="nt",
    sequence=secuencia,
    expect=0.05,
    megablast=True
)
```

Listing 1: Consulta BLASTN Online

2.2. Implementación Local

Para la ejecución local, se creó una base de datos propia (`genomasbase`) usando `makeblastdb`. Dado que las secuencias de prueba eran cortas (< 50 pb), fue necesario ajustar los parámetros del algoritmo local para evitar falsos negativos debidos a filtros de baja complejidad.

Comando optimizado:

```
blastn -task blastn-short -dust no -query query.fasta -db genomasbase
    ...
```

Utilizamos la librería `subprocess` de Python para invocar este comando y capturar la salida en formato XML, la cual fue posteriormente parseada con `NCBIXML`.

Resultados:

- **Online:** Al usar secuencias reales (ej. Insulina), se obtuvieron hits con E-values cercanos a 0,0, confirmando la identificación del gen.
- **Local:** Se logró identificar correctamente secuencias sintéticas introducidas en la base de datos local con 100 % de identidad.

3. Ejercicio 2: BLASTP (Proteínas) y Filtrado

Se implementó una búsqueda de proteínas (`blastp`) filtrando los resultados para conservar solo aquellos con significancia estadística alta.

3.1. Metodología

El script recibe una secuencia de aminoácidos y filtra los alineamientos resultantes (HSPs) que cumplan la condición E-value $< 0{,}001$. Se generó un fichero de salida tabulado con los siguientes campos: ID, Longitud, E-value y Porcentaje de Identidad.

```
if hsp.expect < evalue_umbral:
    identidad = (hsp.identities / hsp.align_length) * 100
    f.write(f"{alignment.hit_id}\t{hsp.expect}\t{identidad:.2f}\n")
```

Listing 2: Cálculo de métricas y filtrado

3.2. Resultados

Al probar con la secuencia de la Hemoglobina humana:

- Se generó correctamente el archivo `blastp_online_filtrado.txt`.
- Los E-values obtenidos fueron del orden de $1e-100$, indicando una homología biológica indiscutible.

4. Ejercicio 3: Filtrado por Organismo

Este ejercicio requirió leer secuencias desde un archivo FASTA y restringir la búsqueda a un organismo específico.

4.1. Estrategia Online (Entrez Query)

En lugar de filtrar los resultados *a posteriori* con Python (lo cual es ineficiente), utilizamos el parámetro `entrez_query` de la API de NCBI para filtrar en el servidor.

```
# Filtrado eficiente en el servidor
filtro = f"{organismo}[ORGN]"
result_handle = NCBIWWW.qblast(..., entrez_query=filtro)
```

Listing 3: Filtro nativo por organismo

Esto garantizó que el 100 % de los resultados devueltos pertenecieran al taxón solicitado (e.g., *Homo sapiens*), optimizando el ancho de banda y el tiempo de procesamiento.

4.2. Estrategia Local

En el entorno local, al carecer de metadatos taxonómicos complejos en nuestra base de datos simple, implementamos un filtro de texto sobre la descripción (`hit_def`) de las secuencias.

5. Ejercicio 4: Suite BLAST Completa

Se evaluaron las cinco variantes principales de la suite BLAST utilizando secuencias reales (Insulina y Hemoglobina) para verificar la coherencia biológica de los resultados.

Herramienta	Consulta vs Base de Datos	Resultado Observado
blastn	ADN vs ADN	Identificación positiva del gen (INS).
blastp	Prot vs Prot	Identificación positiva de la proteína (HBB).
blastx	ADN trad. vs Prot	Traducción correcta; encuentra la proteína.
tblastn	Prot vs ADN trad.	Mapeo inverso; encuentra el gen desde la proteína.
tblastx	ADN trad. vs ADN trad.	Alineamiento más costoso, útil en secuencias diverge

Tabla 1: Resumen de la ejecución de la suite BLAST online.

Observación Crítica: Cuando se ejecutaron estas herramientas con secuencias aleatorias generadas manualmente (."TGC..."), herramientas como `blastx` devolvieron 0 resultados. Esto valida el funcionamiento del algoritmo: al no existir una pauta biológica real ni codificante en una secuencia aleatoria, BLAST correctamente no reporta falsos positivos.

6. Conclusiones

La realización de esta práctica ha permitido consolidar los siguientes conocimientos:

1. **Automatización vs Manual:** El uso de Biopython transforma tareas que tomarían horas en la web en scripts ejecutables en segundos, permitiendo flujos de trabajo reproducibles.

2. **Gestión Local:** La instalación de BLAST+ local es indispensable para trabajar con datos privados o sensibles, aunque requiere una gestión cuidadosa de las bases de datos (`makeblastdb`) y parámetros técnicos (como `-task blastn-short`).

3. **Interpretación de Datos:** Se ha verificado que un E-value bajo es el indicador más fiable de homología. Asimismo, se demostró la importancia de usar filtros nativos (`Entrez`) para optimizar las búsquedas en bases de datos masivas como `nr/nt`.