

# Proyecto Global Alignment by Leon Davis

Este proyecto implementa un alineamiento global de secuencias con manejo de empates y generación de alineamientos óptimos.

---

## Requisitos

- [.NET SDK 7.0 o superior](#)
  - Git
- 

## Cómo ejecutar el proyecto

1. Clonar el repositorio:

```
git clone https://github.com/tu_usuario/tu_repositorio.git
cd tu_repositorio
```

2. Compilar el proyecto:

```
dotnet build
```

3. Ejecutar el proyecto:

```
dotnet run --project GlobalAlignment
```

La salida es por consola indicando el tiempo de ejecución, el score y los alineamientos óptimos, además también se produce un .txt de salida donde se almacena la matriz de alineamiento global (por defecto resultado.txt), la estructura

```

resultado.txt
1  Score final: -3
2
3  Matriz de puntuación:
4  | | - A T A C C A T A C G T
5  -  0 -2 -4 -6 -8 -10 -12 -14 -16 -18 -20 -22
6  T -2 -1 -1 -3 -5 -7 -9 -11 -13 -15 -17 -19
7  A -4 -1 -2  0 -2 -4 -6 -8 -10 -12 -14 -16
8  C -6 -3 -2 -2  1 -1 -3 -5 -7 -9 -11 -13
9  C -8 -5 -4 -3 -1  2  0 -2 -4 -6 -8 -10
0  G -10 -7 -6 -5 -3  0  1 -1 -3 -5 -5 -7
1  A -12 -9 -8 -5 -5 -2  1  0  0 -2 -4 -6
2  T -14 -11 -8 -7 -6 -4 -1  2  0 -1 -3 -3
3
4  Total de óptimos encontrados: 2
5
6  Alineamientos encontrados:
7  Alineamiento 1:
8  -TACC-GA--T
9  ATACCATACGT
0
1  Alineamiento 2:
2  -TACCG-A--T
3  ATACCATACGT

```

del .txt es la siguiente:

## Cómo ejecutar los tests

Para correr los tests unitarios con xUnit:

```
dotnet test
```

## Implementación

Este proyecto implementa un algoritmo de **alineamiento global** entre dos cadenas (secuencias) utilizando programación dinámica, similar al algoritmo de Needleman-Wunsch.

### Explicación del código principal

```

public class GlobalAligner
{
    private const int Match = 1;
    private const int Mismatch = -1;
    private const int Gap = -2;

```

- Se definen constantes para puntajes:

- `Match` = +1 para caracteres iguales.
- `Mismatch` = -1 para caracteres diferentes.
- `Gap` = -2 para espacios (inserciones o borrados).

---

## Método `IsSubstring`

```
public static bool IsSubstring(string main, string sub)
{
    if (main == null || sub == null)
        return false;
    return main.Contains(sub);
}
```

- Método auxiliar que devuelve `true` si `sub` es subcadena de `main`, y `false` en caso contrario.

---

## Método principal `Align`

```
public static (int score, List<(string aligned1, string aligned2)> alignments) Align(string s1,
string s2, string outputPath)
```

- Recibe dos cadenas `s1` y `s2` y un `outputPath` para guardar resultados.
- Devuelve una tupla con el mejor puntaje ( `score` ) y la lista de alineamientos óptimos ( `alignments` ).

---

## Construcción de la matriz de puntajes

- Crea una matriz `scoreMatrix` con tamaño ( `s1.Length+1`, `s2.Length+1` ).
- La primera fila y columna se inicializan con múltiplos de `Gap` para representar la penalización por insertar solo gaps al principio.
- También se mantiene una matriz de `traceback` con listas de direcciones que indican de dónde viene el puntaje máximo en cada celda (Diagonal, Arriba, Izquierda), permitiendo múltiples caminos óptimos.

---

## Rellenar matriz y traceback

- Para cada posición ( `i`, `j` ) se calcula:
  - `scoreDiag` : puntaje si se alinea `s1[i-1]` con `s2[j-1]` (match o mismatch).
  - `scoreUp` : puntaje si hay un gap en `s2` (avance vertical).
  - `scoreLeft` : puntaje si hay un gap en `s1` (avance horizontal).
- Se elige el máximo de estos y se registra en `scoreMatrix[i,j]` .
- Se guardan todas las direcciones que dieron el máximo para permitir múltiples alineamientos óptimos.

---

## Backtracking para recuperar alineamientos

```
private static void Backtrack(string s1, string s2, List<Direction>[,] traceback, int i, int j,
string aligned1, string aligned2, List<(string, string)> alignments)
```

- Recursivamente sigue las direcciones guardadas desde la posición final `(s1.Length, s2.Length)` hasta `(0,0)`.
- En cada paso construye las cadenas alineadas agregando caracteres o guiones ( - ) según la dirección.
- Cuando llega al inicio, agrega el alineamiento completo (invertido, porque se construyó hacia atrás) a la lista `alignments`.

---

### Escritura de resultados

- El resultado final (score y todos los alineamientos óptimos) se escribe en el archivo indicado por `outputPath`.
- También se imprime la matriz de puntajes con etiquetas para facilitar la visualización.

---

### Otras funciones auxiliares

- `Reverse(string s)` : invierte una cadena.
- Enumeración `Direction` : define las direcciones de traceback posibles ( `None` , `Up` , `Left` , `Diagonal` ).

---

Con esta implementación se obtiene el score de alineamiento global entre dos secuencias y todos los alineamientos óptimos que producen ese score.

## Tests

El proyecto incluye una suite de pruebas unitarias usando **Xunit** para verificar el correcto funcionamiento del alineador global.

### Resumen de los tests

- **Test 1:** `Align_ShouldReturnExpectedScoreAndAlignments` Prueba el método `Align` con dos secuencias específicas, verificando que el puntaje devuelto sea el esperado y que uno de los alineamientos óptimos coincida con la solución conocida.
- **Test 2:** `Align_ShouldReturnExpectedScoreAndAlignments2` Similar al primero, con secuencias diferentes y dos alineamientos esperados (acepta cualquiera de ellos), comprobando también el puntaje.
- **Test 3 y 4:** `IsSubstring_ShouldReturnTrue/False` Pruebas para el método `IsSubstring`, verificando que detecte correctamente si una cadena es subcadena de otra o no.

Estos tests aseguran que el algoritmo calcula correctamente el score, encuentra alineamientos óptimos y que la función de búsqueda de subcadenas funciona como se espera.

---

### Ejemplo de salida en consola al ejecutar los tests con Xunit

```
dotnet test
Determining projects to restore...
All projects are up-to-date for restore.
GlobalAlignment -> /home/leon/Documents/UNSA/BIOINFORMATICA/GlobalAlignment/GlobalAlignment/bin/Debug/net8.0/GlobalAlignment.dll
GlobalAlignment.Tests -> /home/leon/Documents/UNSA/BIOINFORMATICA/GlobalAlignment/GlobalAlignment.Tests/bin/Debug/net8.0/GlobalAlignment.Tests.dll
Serie de pruebas para /home/leon/Documents/UNSA/BIOINFORMATICA/GlobalAlignment/GlobalAlignment.Tests/bin/Debug/net8.0/GlobalAlignment.Tests.dll (.NETCoreApp,Version=v8.0)
Herramienta de línea de comandos de ejecución de pruebas de Microsoft(R), versión 17.8.0 (x64)
Copyright (c) Microsoft Corporation. Todos los derechos reservados.

Iniciando la ejecución de pruebas, espere...
1 archivos de prueba en total coincidieron con el patrón especificado.

Correctas! - Con error: 0, Superado: 4, Omitido: 0, Total: 4, Duración: 7 ms - GlobalAlignment.Tests.dll (net8.0)
~/Documents/UNSA/BIOINFORMATICA/GlobalAlignment
```