



UNIVERSIDAD CATÓLICA SAN PABLO

Informe: Algoritmo de Nussinov

Computer Science — Molecular Biology
Computational

Harold Alejandro Villanueva Borda

1. Introducción

Este informe presenta la implementación y análisis del algoritmo de Nussinov-Jacobson para la predicción de estructuras secundarias en ARN, considerando bases independientes. El algoritmo utiliza programación dinámica para encontrar la estructura de menor energía libre posible.

2. Análisis del Algoritmo Principal

2.1. Lógica del Algoritmo

La función principal `calculateMatrix` implementa el algoritmo de Nussinov mediante programación dinámica. El algoritmo sigue los siguientes pasos:

Inicializa una matriz de tamaño $n \times n$ donde n es la longitud de la secuencia. Itera sobre diferentes longitudes de subsecuencias (l). Para cada subsecuencia, considera tres casos:

- Base j no emparejada
- Base i emparejada con base j
- División de la secuencia en dos partes

La fórmula recursiva implementada es:

$$dp[i, j] = \min \begin{cases} dp[i, j - 1] & \text{base } j \text{ libre } dp[i + 1, j - 1] + \alpha(r_i, r_j) \\ i, j \text{ emparejados } \min_{i < k < j} dp[i, k] + dp[k + 1, j] & \text{bifurcación} \end{cases} \quad (1)$$

2.2. Análisis de Complejidad

2.2.1. Complejidad Temporal

El algoritmo tiene una complejidad temporal de $O(n^3)$ debido a:

- Bucle exterior sobre longitudes: $O(n)$
- Bucle sobre posiciones iniciales: $O(n)$
- Bucle interno para bifurcaciones: $O(n)$

2.2.2. Complejidad Espacial

La complejidad espacial es $O(n^2)$ debido a la matriz de programación dinámica.

3. Experimentos y Resultados

3.1. Análisis de Secuencias de Prueba

Para la secuencia GGAAAUCC:

- Con modelo de energía complejo:
 - Estructura: $(((.)))$
 - Energía final: -14
 - Pares formados: G0-C7, G1-C6, A2-U5
- Con modelo de energía simple:
 - Estructura:
 - Energía final: -3

Para la secuencia ACUCGAUCCGAG:

- Estructura: $.((((().).)))$
- Energía final: -23
- Pares formados: C1-G12, U2-A11, C3-G10, G4-C8, A5-U6

4. Respuestas a Preguntas de Investigación

4.1. Relación entre Predicción de Estructuras ARN y Palíndromos

La predicción de estructuras secundarias de ARN está estrechamente relacionada con la búsqueda de palíndromos porque:

- Las estructuras de ARN forman "palíndromos biológicos" donde las bases complementarias se emparejan.
- La estructura en horquilla (stem-loop) es esencialmente un palíndromo con complementariedad de bases.
- Al igual que en los palíndromos, la lectura en dirección opuesta (considerando complementariedad) produce la misma estructura.

4.2. Efecto de la Energía Libre en los Resultados

Los resultados muestran que:

- El modelo de energía complejo (-5 CG, -4 AU, -1 GU) produce estructuras más estables y complejas.
- El modelo simple (-1 para todos los pares) tiende a producir estructuras más simples o ninguna estructura.
- La diferencia en los resultados demuestra la importancia de una función de energía precisa para predicciones biológicamente relevantes.

4.3. Logros y Mejoras del Algoritmo

Las mejoras más relevantes en el campo incluyen:

- Incorporación de parámetros termodinámicos más precisos
- Consideración de pseudonudos y otras estructuras complejas
- Integración con métodos de aprendizaje automático
- La mejora más significativa ha sido la incorporación de parámetros termodinámicos experimentales, que ha aumentado significativamente la precisión de las predicciones.

5. Conclusiones

- El algoritmo de Nussinov-Jacobson proporciona una base sólida para la predicción de estructuras secundarias de ARN.
- La elección del modelo de energía afecta significativamente los resultados de la predicción.
- La complejidad temporal $O(n^3)$ limita su aplicación a secuencias relativamente cortas.
- Las mejoras modernas en los parámetros energéticos y la consideración de estructuras más complejas han aumentado significativamente la utilidad del algoritmo en aplicaciones biológicas reales.
- La implementación demuestra la importancia de la programación dinámica en la resolución de problemas de biología computacional.

6. Implementación en Github

El código fuente del análisis se encuentra disponible en GitHub: [GitHub](#).