

## Práctica 4 – Plegado de cadenas de ARN

### Definición del problema y objetivo

El objetivo de esta práctica es estudiar el algoritmo de Nussinov que resuelve el problema del plegado de secuencias de ARN utilizando la técnica de programación dinámica. En el documento folding.pdf (disponible en el campus virtual) puedes encontrar la definición del problema y el algoritmo.

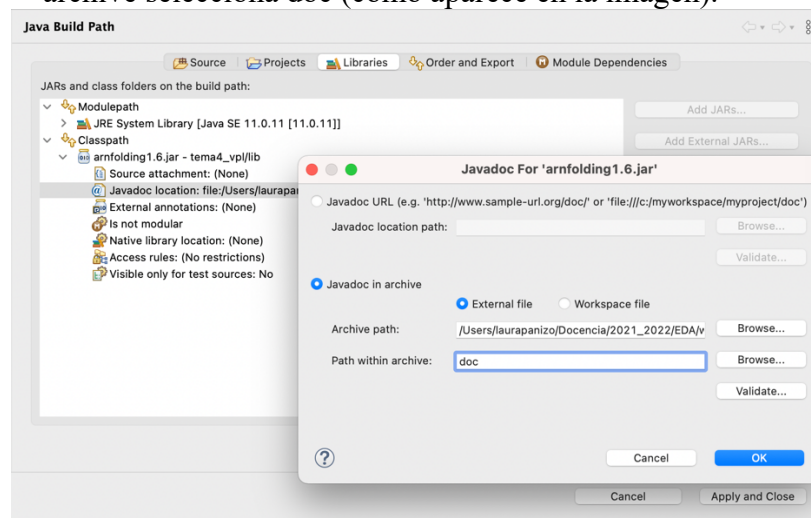
### Realización de la práctica

El estudiante tiene que implementar los métodos `_run` y `reconstructSolution` de la clase `Nussinov`. El método `reconstructSolution` recibe como parámetro un array bidimensional que representa la matriz de decisiones `D` y los índices `i` y `j` desde los que hay reconstruir la solución y devuelve la secuencia de ARN como un `String` utilizando la notación Dot-Bracket (transparencia 15 del documento folding.pdf).

El método `_run` recibe como parámetros un `String` con la cadena de ARN y el tamaño mínimo de plegado. En este método hay que implementar la ecuación del Bellman (transparencias 12-14 del documento folding.pdf) y devolver la secuencia de ARN plegada, es decir, que este método debe calcular la matriz `D` de decisiones y hacer una llamada al método `reconstructSolution` para obtener la secuencia de ARN plegada.

Además de esta clase, se proporcionan las siguientes clases y librerías:

- `arn1.6.jar` y `arn1.6-doc.jar`: esta librería contiene algunas clases necesarias para la ejecución del programa (no hay que utilizar ninguna para la implementación de los métodos que se piden). Para importar la librería en el proyecto eclipse:
  - Copiar los ficheros jar en el proyecto. Por ejemplo, puedes crear una carpeta `lib` dentro del proyecto para mantener los ficheros organizados.
  - Sobre el nombre del proyecto, pulsa el botón secundario y selecciona `Configure Build Path > Pestaña Libraries`
  - Si en el área central te aparecen `Module Path` y `Class Path` selecciona `Class Path`
  - Pulsa el botón `Add JARs` y selecciona el fichero `arn1.6.jar`
  - Para añadir la documentación asociada despliega la librería asociada al jar, y haz doble click sobre `Javadoc location`. Selecciona `Javadoc in archive > External file`. En el path añade el fichero `arn1.6-doc.jar`, y en `Path within archive` selecciona `doc` (como aparece en la imagen).



- TestRNAFolding: en esta clase se encuentra en método main (no hay que modificarlo). El método main recibe en los argumentos de entrada el modo de ejecución, que puede ser uno de los siguientes:
  - o -s <secuencia> <minlop>: el algoritmo pliega la *secuencia* plegando nucleótidos que estén al menos separados *minlop* posiciones.
  - o -f <input-file> <minlop>: El fichero *input-file* contiene en cada línea una secuencia de ARN. Con el modo -f el algoritmo calcula el plegado de cada una de las secuencias usando *minlop* como separación mínima para plegar los nucleótidos.
  - o -r <init\_length> <doubling> <num\_seq> <minlop>: en este modo se generan secuencias de ARN aleatorias con diferentes longitudes. Se generarán *num\_seq* secuencias de cada longitud, que empezará con *init\_length* y se irá doblando la tantas veces como indique el parámetro *doubling*. Por último, se pasa como parámetro la separación mínima para plegar nucleótidos. El programa genera un fichero con los tiempos de ejecución que se puede utilizar para realizar un estudio experimental de la complejidad del algoritmo. Si utilizamos la opción -r 10 5 20 6 el programa generará de forma aleatoria 20 secuencias de longitud 10, 20 de longitud 20, 20 de longitud 40, y así hasta generar secuencias de longitud 320.
- Fichero de prueba (seqX.txt) para usar con el modo -f y la correspondiente salida (seqX.out).
- analyze.R: este fichero contiene un script que procesa los ficheros que se generan con el modo -r. Hay una guía en el campus que explica como instalar el entorno R y ejecutar el script.

## Entrega y evaluación

La evaluación de la práctica consiste en la entrega y evaluación automática del código del alumno y una memoria (de no más de 3 páginas) en la que el alumno presenta los objetivos, el entorno de ejecución donde se ha realizado las pruebas, los resultados teóricos esperados y los resultados que ha obtenido en el estudio experimental de la complejidad del algoritmo. A continuación, se detalla la nota que se puede obtener:

Nota	Descripción
0	No pasa el test en laboratorio ni entrega memoria o el sistema detecta plagio en el código o la memoria
[1,4]	No pasa el test en laboratorio. Entrega la memoria y pasa los test fuera de plazo
[4,6]	Pasa parcialmente los test de laboratorio. Entrega memoria y pasa test fuera de plazo
[6,10]	Pasa los test en laboratorio y entrega memoria

Para la evaluación automática del código el estudiante debe subir el fichero Nussinov.java a una actividad del campus que realizará una serie de pruebas sobre la implementación. El código hay que subirlo siempre al final de la sesión de laboratorio correspondiente, y si no se pasan todas las pruebas tendrá una segunda oportunidad fuera de plazo. La fecha se comunicará al final de la sesión de laboratorio.