Comparación y análisis de secuencias de ADN genómico

Julián Quintero, Natalia Monroy Rosas, Santiago Rodríguez Camargo,
Víctor Manuel Torres Alonso

No. de Equipo Trabajo: 2

INTRODUCCIÓN

Este documento es una aproximación al estudio de las semejanzas y diferencias entre secuencias de ADN genómico de diferentes organismos, aplicando estructuras de datos lineales para el almacenamiento, búsqueda y ordenamiento de múltiples sub-secuencias de ADN.

I. DESCRIPCIÓN DEL PROBLEMA A RESOLVER

Uno de los procesos más efectivos para comprender el genoma se realiza en genómica comparativa, y consiste en identificar secuencias de ADN para establecer regiones de cadenas similares y diferentes en diferentes organismos. Las herramientas tecnológicas de secuenciación y comparación de ADN han sido fundamentales en el campo de la genómica, por ejemplo, en el desarrollo del Proyecto del Genoma Humano. La secuenciación es una tarea de alta complejidad, sin embargo, una vez codificado el genoma es posible hacer diversas comparaciones para encontrar genes que determinan una característica específica. El proyecto se centrará en determinar la estructura de datos y el algoritmo más eficiente para realizar este tipo de comparación.

II. USUARIOS DEL PRODUCTO DE SOFTWARE

El producto está diseñado para ser utilizado por genetistas y bioquímicos que necesiten analizar y comparar secuencias de ADN para diferentes organismos.

III. REQUERIMIENTOS FUNCIONALES DEL SOFTWARE

Cada funcionalidad se debe especificar así:

- *Comparación de secuencias de ADN :*

  El programa determina cuántas y cuáles subcadenas de longitud *m,* tienen en común dos secuencias de ADN.
  - ❖ *Acciones iniciadoras y comportamiento esperado*: El usuario selecciona el par de secuencias a comparar a partir de las opciones que observa en pantalla. Luego introduce la longitud de subcadenas para realizar la comparación. El programa retorna una lista con las subcadenas comunes a ambas secuencias e indica cuántas son.

- *Subcadena(s) más frecuentes en una secuencia:*

  El programa encuentra cuáles son las subcadenas de longitud *m* más frecuentes en una secuencia de ADN.
  - ❖ *Acciones iniciadoras y comportamiento esperado*: El usuario selecciona la secuencia a analizar a partir de las opciones en pantalla. Luego introduce la longitud de subcadena para realizar la operación. El programa retorna la o las subcadenas que aparecen mayor cantidad de veces en la secuencia para la longitud ingresada.

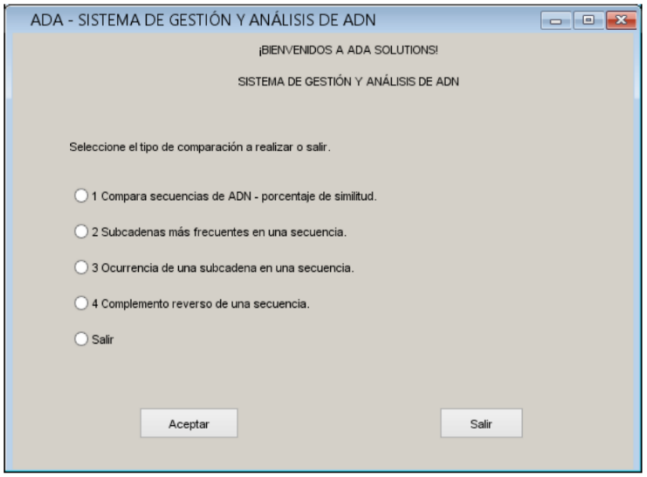
- *Ocurrencia de una subcadena una secuencia:*

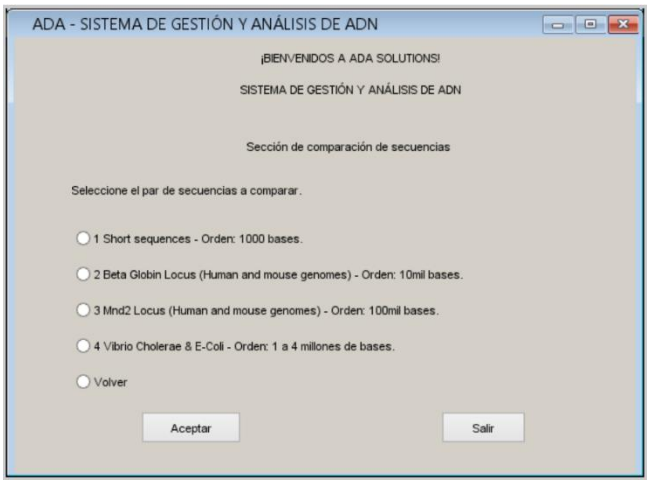
  Se determinan los índices donde ocurre una subcadena en una subsecuencia.
  - ❖ *Acciones iniciadoras y comportamiento esperado*: El usuario selecciona la secuencia que desea a partir de las opciones en pantalla. Luego introduce la subcadenas para la búsqueda de los índices. El programa retorna una lista con los índices dónde aparece la subsecuencia ingresada en la cadena seleccionada

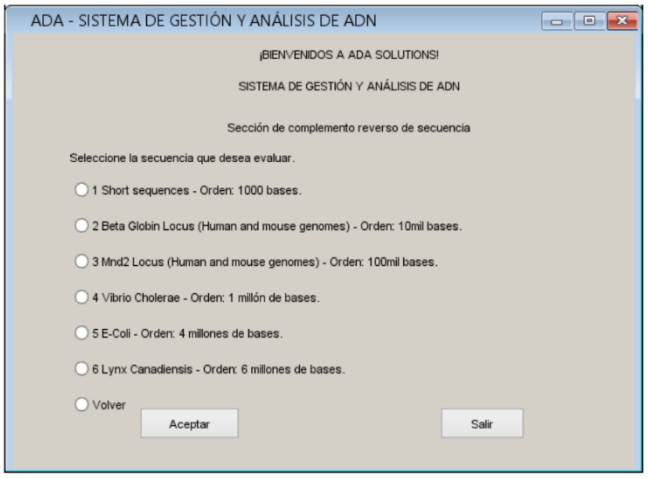
- *Complemento reverso de una secuencia:*

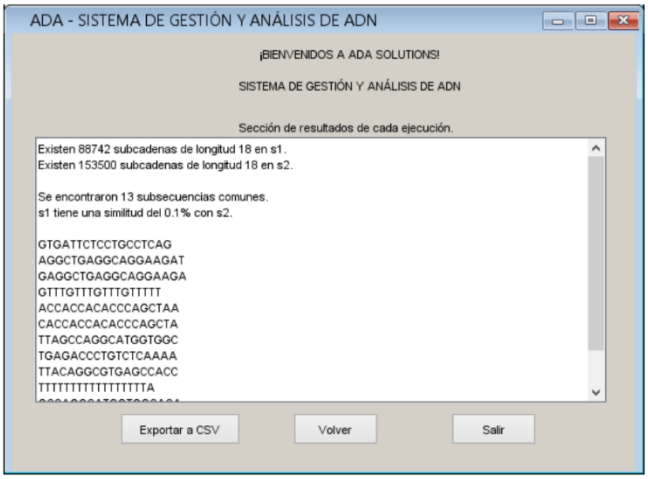
  Retorna el complemento reverso de una secuencia de ADN.
  - ❖ *Acciones iniciadoras y comportamiento esperado*: El usuario selecciona la secuencia que desea a partir de las opciones en pantalla. El programa retorna la cadena correspondiente al complemento reverso.

IV. DESCRIPCIÓN DE LA INTERFAZ DE USUARIO PRELIMINAR









V. ENTORNOS DE DESARROLLO Y DE OPERACIÓN

ADA - Sistema de gestión y análisis de ADN se está desarrollando en lenguaje JAVA puro, realizando la edición en los IDE's Eclipse e IntelliJ IDEA.

Dados los beneficios del aspecto multiplataforma de los desarrollos construidos en JAVA, ADA en principio no tiene limitación, no obstante ya se está ejecutando en ordenadores con plataformas Windows en sus versiones 8 y 10, en hardware con procesadores de 4 núcleos y memoria RAM de 4 a 8 GB.

VI. PROTOTIPO DE SOFTWARE INICIAL

El primer prototipo del software ADA se encuentra en la plataforma web GitHub:
https://github.com/EDGenomica/ADN_comp_proy

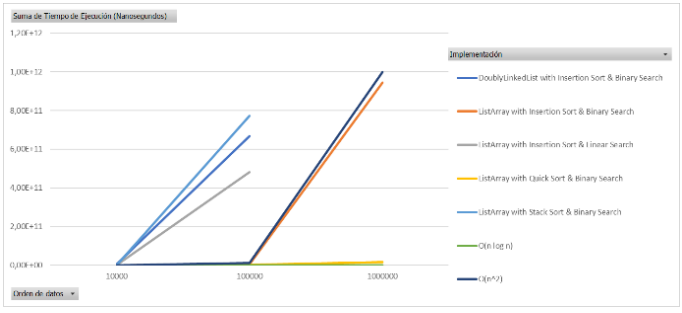
VII. IMPLEMENTACIÓN Y APLICACIÓN DE LAS ESTRUCTURAS DE DATOS

- *Arreglos:* Se implementó una lista con arreglos para almacenar las subsecuencias de una longitud determinada de una cadena. A esta lista se le realizan operaciones de búsqueda, inserción, eliminación y ordenamiento.
- *Linked List:* De igual forma, para efectos de comparación, se implementó esta estructura para almacenar las subcadenas de una secuencia. Se realizan operaciones de búsqueda, inserción, eliminación y ordenamiento.
- *Pilas:* Se implementó una pila con referencias para almacenar las subcadenas resultantes de las funcionalidades de comparación, subcadenas más frecuentes. También se utiliza para obtener el complemento reverso de una secuencia.
- *Colas:* Se creó una cola con referencias para almacenar los índices en los que aparece una subcadena dentro de una secuencia.

VIII. PRUEBAS DEL PROTOTIPO Y ANÁLISIS COMPARATIVO

- *Comparación de secuencias de ADN:*
  Para la funcionalidad principal del programa se realizaron diferentes implementaciones de estructuras de datos y algoritmos:

COMPARACIÓN DE SECUENCIAS				
Implementación	Tiempo de ejecución			Big (O)
	10 mil	100 mil	1 millón	
DoublyLinkedList with Insertion Sort & Binary Search	6,98E+09	6,67E+11		O(n^2)
ListArray with Insertion Sort & Binary Search	7,42E+07	4,84E+09	9,44E+11	O(n^2)
ListArray with Insertion Sort & Linear Search	1,73E+09	4,80E+11		O(n^2)
ListArray with Quick Sort & Binary Search	2,11E+07	2,19E+08	1,64E+10	O(n log n)
ListArray with Stack Sort & Binary Search	2,06E+09	7,73E+11		O(n^2)

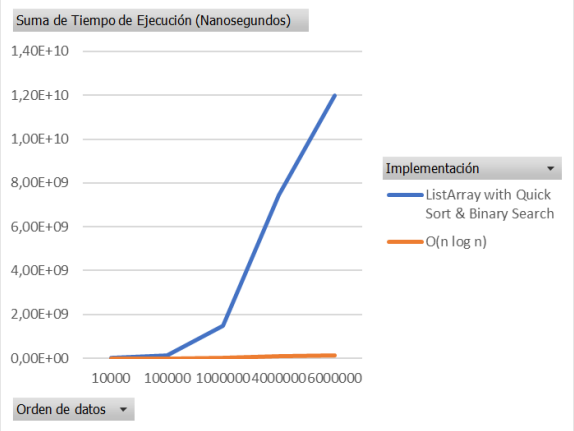


Como se puede observar, el conjunto de estructura lineal y algoritmo, más óptimo para la comparación de secuencias es una *Lista implementada con arreglos*, ordenada con *Quick Sort* y cuya búsqueda se realiza

con un algoritmo *Binary Search*. Esta implementación se utilizará para las otras funcionalidades.

- *Subcadena(s) más frecuentes en una secuencia*

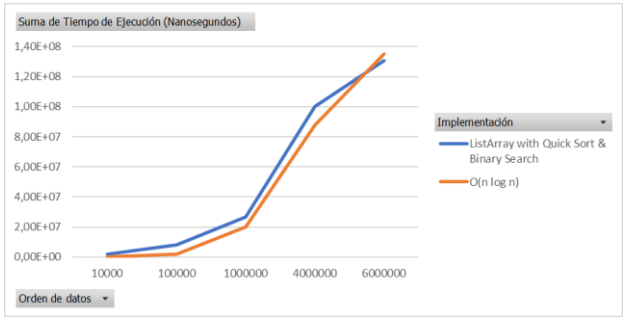
SUBSTRING MÁS FRECUENTE		
Orden de datos	Tiempo de ejecución	Big (O)
10mil	1,93E+07	
100mil	1,39E+08	> O(n log n)
1millón	1,49E+09	< O(n^2)
4millones	7,43E+09	
6millones	1,20E+10	



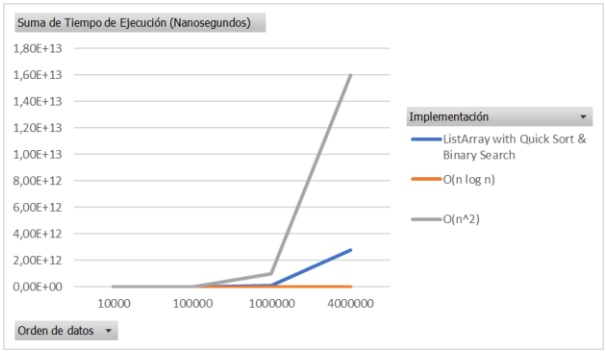
- *Ocurrencia de una subcadena una secuencia*

OCCURRENCIA SUBSTRING EN SECUENCIA		
Orden de datos	Tiempo de ejecución	Big (O)
10mil	1,67E+06	
100mil	8,13E+06	
1millón	2,66E+07	O(n log n)
4millones	9,99E+07	
6millones	1,20E+10	

- *Complemento reverso de una secuencia*



COMPLEMENTO REVERSO DE SECUENCIA		
Orden de datos	Tiempo de ejecución	Big (O)
10mil	3,31E+07	
100mil	1,28E+09	> O(n log n)
1millón	1,25E+11	< O(n^2)
4millones	2,79E+12	



IX. ROLES Y ACTIVIDADES

Integrante	Rol(es)	Actividades fundamentales
Julián Quintero	Investigador, Técnico	Consultar a los otros miembros del equipo, atento que la información sea constante para todos. Aportar con la organización y plan de trabajo. Aporta técnicamente en el desarrollo del proyecto.
Natalia Monroy	Lideresa, Investigadora, Técnica	Aportar con la organización y plan de trabajo. Aporta técnicamente en el desarrollo del proyecto. Mantener el contacto entre todos. Líder técnico que propende por coordinar las funciones y actividades operativas.
Santiago Rodríguez	Lider, Experto, Técnico	Aportar con la organización y plan de trabajo. Aporta técnicamente en el desarrollo del proyecto. Mantener el contacto entre todos. Líder técnico que propende por coordinar las funciones y actividades operativas.
Víctor Torres	Investigador, Técnico	Consultar otras fuentes. Propender por resolver inquietudes comunes para todo el equipo. Aportar con la organización y plan de trabajo. Aporta técnicamente en el desarrollo del proyecto.

X. DIFICULTADES Y LECCIONES APRENDIDAS

- Las estructuras de datos lineales comprenden un extenso conjunto de conceptos, lo que hizo que resultara dispendioso encontrar entre todas las posibles implementaciones, las estrictamente necesarias y adecuadas para el procesamiento de los datos.
- El campo de la genómica comparativa sigue en pleno auge por lo cual hay información investigativa muy diversa, manejada desde distintas ramas del conocimiento, lo que pudo ser un reto al momento de definir los conjuntos de datos apropiados para aplicarles estructuras de datos lineales, pues hay bases de datos muy completas, pero con grados de procesamiento que impiden nuestro objetivo principal.
- En cuanto al requisito de poder procesar data con volúmenes del orden de millones de datos individuales, el hardware del que disponemos para ello, que consiste normalmente en computadores de línea de consumo

para usuarios finales, ha resultado ser una enorme limitación, tomando tiempos largos del orden de horas.

- La principal lección ha sido observar en vivo lo interesante de estas estructuras de datos lineales que hemos usado en pequeños ejercicios de clase y en talleres, ahora procesando volúmenes de datos quizá muy parecidos a la vida real.

XI. REFERENCIAS BIBLIOGRÁFICAS

[1] J. Sandeep. et all, “GeeksforGeeks quickSort,” Accessed october 2020, [Online]. Available: https://www.geeksforgeeks.org/quick-sort/?ref=lbp

[2] J. Sandeep. et all, “GeeksforGeeks Binary Search,” Accessed october 2020, [Online]. Available: dhttps://www.geeksforgeeks.org/binary-search/.

[3] Kent WJ, Sugnet CW, Furey TS, Roskin KM, Pringle TH, Zahler AM, Haussler D. “The human genome browser at UCSC API REST”. Genome Res. 2002 Jun;12(6):996-1006, [Online]. Available: https://genome.ucsc.edu/goldenPath/help/api.html#Return

[4] (2020) fasta. Accessed october 2020, [Online]. Available: http://ftp.ensembl.org/pub/release-101/fasta/

[5] N. Andrew, “Finding Hidden Messages in DNA (Bioinformatics I),” October 18 2020. [Online]. Available: https://www.coursera.org/learn/dna-analysis

[6] P. Compeau, et all, “Bioinformatics algorithms,” Where in the Genome Does DNA Replication Begin?, [Online]. Available: https://www.bioinformaticsalgorithms.org/bioinformatics-chapter-1

[7] National Center for Biotechnology Information, U.S. National Library of Medicine, “Home - Genome - NCBI,” Ncbi.nlm.nih.gov, 2020, [Online]. Available: https://www.ncbi.nlm.nih.gov/genome

[8] S.A. Goldman and K. J. Goldman, “A Practical Guide to Data Structures and Algorithms Using Java,” [Online]. Available: http://goldman.cse.wustl.edu/crc2007/projects/

[9] J.T. Streib and T. Soma, *Guide to Data Structures A Concise Introduction Using Java*, Cha, Switzerland, Springer International Publishing AG, eBook, 2017, Chapter 1-4