UNIVERSIDAD DE CHILE
FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS
DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN

Mapeo de secuencias de ADN a árboles filogenéticos usando compresión de gramáticas

PROPUESTA DE TEMA DE MEMORIA PARA OPTAR AL TÍTULO DE INGENIERO CIVIL EN COMPUTACIÓN

Victor Alfaro

MODALIDAD:
Memoria

PROFESOR GUÍA:
Gonzalo Navarro

SUPERVISOR:
Gonzalo Navarro

SANTIAGO DE CHILE
2025

1. Introducción

La bioinformática es una disciplina científica que combina biología, informática, matemáticas y estadística con el fin de recopilar y analizar grandes volúmenes de datos biológicos como secuencias de ADN, aminoácidos, entre otros. Uno de sus objetivos principales es comprender cómo las secuencias genómicas ("texto"del ADN) se han ido presentando en la evolución de las especies. En este contexto entran en juego los árboles filogenéticos, estructura que permite representar las relaciones de parentesco entre especies y que muestra los cambios en los genes con el pasar del tiempo. Teniendo en cuenta la gran cantidad de especies que existe y que además se tienen una enorme cantidad de colecciones de genomas (conjunto de todos los genes de un ser vivo), es un gran desafío contestar preguntas de interés como por ejemplo: ¿Dónde aparece una secuencia particular dentro de un genoma?, ¿Cuántas veces aparece una secuencia en un conjunto de genomas?, etc.

El problema detalladamente consiste en que, dada una colección de genomas organizada según el árbol filogenético, se busca una secuencia de ADN o un gen ,se identifica todas sus ocurrencias en la colección, se mapean a los genomas que las contienen y se calcula el ancestro común más bajo (Lowest Common Ancestor-LCA) de esos nodos, el cual, es un candidato para la especie en la cual comenzó a expresarse.

Este problema es complejo por el hecho de que las secuencias no se suelen encontrar exactamente debido a mutaciones y otros fenómenos. Por lo tanto, no basta buscar ocurrencias exactas, sino que se buscan las coincidencias máximas, representadas por una estructura llamada maximal exact matches (MEMs). También, como se mencionó, las colecciones de genomas tienen volúmenes muy grandes, por lo que la eficiencia contempla el tiempo de búsqueda y también la memoria utilizada.

En el último tiempo, se han desarrollado diversas soluciones, los llamados índices comprimidos, los cuales son capaces de manejar colecciones muy repetitivas. El r-index y las estructuras que emplean compresión de gramáticas son las más destacadas, ya que logran reducir el espacio ocupado manteniendo tiempos de búsqueda eficientes. Entre las estructuras de datos clásicas, el arreglo de sufijos (SA) juega un rol central: este arreglo almacena todas las posiciones iniciales de los sufijos de un texto en orden lexicográfico, lo que permite localizar patrones mediante búsquedas binarias sobre él. Una variante directa es el arreglo diferencial de sufijos (DSA), que en lugar de guardar los valores completos de SA, almacena las diferencias entre posiciones consecutivas. En colecciones genómicas, estas diferencias suelen ser muy repetitivas, lo que hace al DSA altamente compresible. El DSA es crucial para encontrar el ancestro común, ya que dado un patrón (o un MEM) el FM-index (índice comprimido que devuelve el rango en donde se encuentra un patrón en particular) entrega su intervalo [sp..ep] en SA. Para mapear ese MEM a especies necesitamos las ocurrencias más a la izquierda y más a la derecha en el texto concatenado por el orden filogenético (min y max de SA[sp..ep]);los cuales serían los genomas extremos, y su LCA es el candidato a “primera aparición”. Guardar SA plano para hacer ese min/max es costoso; en cambio, comprimir el DSA con gramáticas y entregar metadatos a cada regla con suma, mínimo y máximo de prefijos permite responder RMQ (min/max sobre SA) en poco espacio, haciendo viable el cálculo del LCA a gran escala.

Sin embargo, no existe hasta ahora una implementación práctica de un índice comprimido

que trabaje directamente sobre el DSA y que soporte de manera eficiente las operaciones necesarias para construir tablas de MEMs.

Tener una solución a este problema representaría un gran aporte para la bioinformática y la filogenia. Una estructura que combine alta compresión con tiempos de búsquedas eficientes permitiría procesar colecciones de genomas enormes, facilitando la tarea de las otras áreas de la ciencia. La propuesta de implementar un índice basado en la gramática del DSA es algo poco explorado en la literatura, por lo que tiene potencial de ofrecer mejoras en comparación a otras implementaciones.

2. Situación Actual

La comunidad de algoritmos de texto y bioinformática ha ofrecido diversas soluciones para el problema de indexar grandes colecciones de genomas. En particular, el desarrollo de índices comprimidos ha permitido trabajar con volúmenes masivos de datos aprovechando la alta repetitividad entre secuencias de especies cercanas.

Una de las primeras investigaciones se centró en los árboles y arreglos de sufijos comprimidos(CST,CSA) [1] . Estos permiten realizar búsquedas exactas de patrones con espacio menor que las estructuras clásicas. Se demostró que era posible responder consultas de ancestros en espacio comprimido. Sin embargo, estas estructuras tienen limitaciones para construir por ejemplo la tabla de MEMs; en temas de espacio sigue siendo elevado para colecciones muy grandes.

Con un SA tradicional, el problema se resolvería buscando el patrón mediante búsqueda binaria, lo que entrega un intervalo [sp..ep] en el SA con todas sus ocurrencias. Para un MEM, es necesario identificar las ocurrencias más a la izquierda y a la derecha, es decir, calcular min(SA[sp..ep]) y max(SA[sp..ep]). Sin embargo, con un SA plano este cálculo requiere recorrer todo el rango, con un costo proporcional a su longitud (ep - sp + 1), lo que resulta ineficiente cuando los MEMs son muy frecuentes en colecciones grandes.El r-index [2] [6], estructura basada en la Transformada de Burrows–Wheeler (BWT), representó un avance fundamental. Este índice se apoya en el parámetro r, definido como el número de runs en la BWT, es decir, la cantidad de subcadenas máximas de caracteres idénticos consecutivos en la transformada. En colecciones genómicas altamente repetitivas, el valor de r suele ser mucho menor que la longitud total del texto (n), lo que permite representar el índice en espacio proporcional a r en lugar de n. Gracias a esta propiedad, el r-index soporta operaciones de conteo y localización de patrones de manera muy eficiente en escenarios con alta repetitividad. Sin embargo, cuando la repetitividad es intermedia, el valor de r crece y el espacio requerido se vuelve considerable; además, su implementación debe complementarse con estructuras adicionales para responder consultas de tipo RMQ (Range Minimum Query). Esta limitación motiva la propuesta de utilizar el arreglo diferencial de sufijos (DSA) comprimido mediante gramáticas, lo que permitiría calcular min y max-SA en poco espacio, reemplazando el SA explícito y mejorando así el costo de estas operaciones.

También se ha explorado el uso de compresión por gramáticas. En 2019 se mostró cómo aplicar gramáticas sobre el arreglo diferencial de LCP (Longest Common Prefix) [5], que es un arreglo donde cada posición indica la longitud del prefijo común más largo entre sufijos

consecutivos en el SA. Esta estructura es fundamental porque permite responder consultas de coincidencia y extensión de patrones en árboles de sufijos. El enfoque de comprimir el LCP diferencial con gramáticas demostró que es posible construir índices muy compactos con buen rendimiento en consultas. Sin embargo, aún no existe una implementación equivalente sobre el arreglo diferencial de sufijos (DSA). Dado que el DSA captura de forma natural las variaciones locales en el SA y tiende a ser muy repetitivo en colecciones genómicas, su compresión mediante gramáticas aparece como una oportunidad para obtener índices más pequeños sin perder la capacidad de responder consultas como RMQ. [4].

En el contexto de la clasificación taxonómica, herramientas como Kraken utilizan k-mers, que son todas las subcadenas de longitud fija k de un genoma, para indexar colecciones y asignar lecturas a subárboles en un árbol filogenético [3]. Sin embargo, en trabajos recientes se ha mostrado que emplear MEMs en lugar de k-mers puede mejorar significativamente la precisión de la clasificación. Este enfoque requiere índices más potentes, ya que los MEMs son más costosos de calcular que los k-mers. Para abordar este desafío se han propuesto aproximaciones basadas en representaciones comprimidas, como KATKA kernels y minimizer digests, que reducen el espacio de almacenamiento a costa de introducir falsos positivos. Estas soluciones evidencian la necesidad de desarrollar nuevos índices comprimidos que logren un mejor equilibrio entre espacio, tiempo de búsqueda y exactitud.

Si bien existen avances significativos, falta una solución que combine la capacidad de responder operaciones complejas, el uso directo de DSA y una integración practica en problemas filogenéticos.

3. Objetivos

Objetivo General

Diseñar, implementar y evaluar un índice comprimido basado en gramáticas sobre el arreglo diferencial de sufijos (DSA), capaz de responder consultas fundamentales de tipo RMQ de manera eficiente en colecciones genómicas altamente repetitivas, con el fin de mejorar la detección de ocurrencias relevantes de secuencias y apoyar la identificación del nodo ancestral de aparición de genes en árboles filogenéticos.

Objetivos Específicos

1. **Revisar el estado del arte en índices comprimidos para colecciones repetitivas**: Teniendo los principales focos actuales (FM-index, r-index, sr-index, KATKA kernels, entre otros), se identificarán sus ventajas, limitaciones y aplicabilidad al problema de construir tablas de MEMs y consultas de LCA.
2. **Definir formalmente la estructura de datos propuesta sobre el DSA**: Se diseñará un esquema de compresión basado en gramáticas para el arreglo diferencial de sufijos, incorporando en cada no terminal la información necesaria (suma total, mínimo prefijo, máximo prefijo) para soportar consultas de rango.
3. **Implementar un prototipo funcional del índice DSA-gramática**: Se implementará el DSA, su compresión por gramática y las operaciones de consulta requeridas.

4. **Integrar el índice en el pipeline de detección de MEMs**: El prototipo se conectará con un algoritmo de búsqueda de MEMs sobre un FM-index aumentado, reemplazando el acceso al SA por el índice de la gramática del DSA para calcular las ocurrencias más izquierda y más derecha.
5. **Evaluar experimentalmente el rendimiento del índice**: Se diseñarán y ejecutarán experimentos comparativos sobre colecciones de genomas reales y sintéticas, midiendo espacio ocupado, tiempo de construcción y tiempos de consulta, y también la tasa de aciertos en la detección de MEMs. También se comparará con el baseline que usa el r-index en términos de todas las métricas ya mencionadas.
6. **Analizar los resultados** Evaluar en qué grado se mejora la eficiencia espacio/tiempo respecto a las soluciones existentes.

4. Solución Propuesta

La solución propuesta consiste en diseñar e implementar un índice comprimido sobre el arreglo diferencial de sufijos (DSA), utilizando compresión de gramáticas para representar de manera compacta las diferencias entre posiciones del arreglo de sufijos. Se usará un algoritmo de compresión tipo Repair u otro generador de gramáticas, al cual se extenderá con información adicional en cada no terminal (suma total, mínimo prefijo y máximo prefijo de su expansión). Estos valores permitirán responder consultas como RMQ directamente sobre el DSA comprimido, sin expandir el arreglo completo.

Los inputs, como en otros trabajos de investigación, serán colecciones de genomas concatenados por el símbolo $ para representar los limites entre cada uno. También, se tendrá la implementación del FM-index aumentado, que permitirá la búsqueda de MEMs y operaciones sobre BWT, SA y LCP. La función que cumplirá el DSA-gramática será reemplazar el acceso directo al SA.

Para la implementación se usará C++ o C, ya que estos lenguajes son eficientes para estructuras de datos de bajo nivel. A su vez, la comparación y evaluación se hará incluyendo los baselines existentes, como el FM-index aumentado, el r-index, sr-index, entre otros.

Evaluación

La evaluación del trabajo está muy ligada con la parte experimental. Se harán experimentos sobre colecciones genómicas sintéticas y reales, las cuales se permitan variar su repetitividad y el tamaño. Como baselines, se usarán estructuras ya implementadas, como el FM-index aumentado (augmented), r-index, entre otros.

Las métricas que se ocuparán incluyen el espacio ocupado (bit per character-bpc), tiempo de construcción del indice y tiempo de consulta.

5. Plan de Trabajo (Preliminar)

1. **Revisión del estado del arte**: Estudiar en profundidad los artículos relacionados con los índices comprimidos. (2 semanas)

2. **Formalización del problema y diseño de estructura** : Diseñar la estructura gramatical sobre el DSA. (4 semanas)
3. **Implementación del prototipo del índice DSA-gramática**: (4 semanas)
4. **Integración con el pipeline de búsqueda de MEMs** (2 semanas)
5. **Evaluación experimental** (1 semana)
6. **Redacción de informe final** (2 semanas)

Referencias

[1] Abeliuk, Andrés, Rodrigo Cánovas y Gonzalo Navarro: *Practical Compressed Suffix Trees*. Algorithmica, 65(3):606–635, 2013.

[2] Cobas, Dustin, Travis Gagie y Gonzalo Navarro: *Fast and Small Subsampled R-indexes*. ACM Transactions on Algorithms, 21(2):1–32, 2025.

[3] Draesslerová, Dominika, Omar Ahmed, Travis Gagie, Jan Holub, Ben Langmead, Giovanni Manzini y Gonzalo Navarro: *Taxonomic Classification with Maximal Exact Matches in KATKA Kernels and Minimizer Digests*. En *Proceedings of the 22nd International Symposium on Experimental Algorithms (SEA 2024)*, LIPIcs, páginas 10:1–10:13. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2024.

[4] Fischer, Johannes y Volker Heun: *Theoretical and Practical Improvements on the RMQ-Problem, with Applications to LCA and LCE*. En *Journal of Discrete Algorithms*, volumen 34, páginas 3–15, 2016.

[5] Gagie, Travis, Gonzalo Navarro y Nicola Prezza: *Fully-Functional Suffix Trees and Optimal Text Searching in BWT-runs Bounded Space*. Journal of the ACM, 66(3):1–54, 2019.

[6] Gagie, Travis, Gonzalo Navarro y Nicola Prezza: *Fully Functional Suffix Trees and Optimal Text Searching in BWT-runs Bounded Space*. Journal of the ACM, 67(1):1–54, 2020.