Universidad Zaragoza
1542

Trabajo Fin de Máster

ENSAMBLAJE DE GENOMA USANDO TÉCNICAS DE OPTIMIZACIÓN CUÁNTICAS

GENOME ASSEMBLY USING QUANTUM OPTIMIZATION TECHNIQUES

Autor/es

Pilar Mollá Ortigas

Director/es

David González Rojas

Vanessa Bueno Sancho

MÁSTER INGENIERÍA BIOMÉDICA

ESCUELA DE INGENIERÍA Y ARQUITECTURA

Agosto 2023

AGRADECIMIENTOS

A la Universidad de Zaragoza por brindarme durante todos estos años un entorno propicio y una enseñanza fundamental para mi desarrollo profesional y personal.

A la empresa NTT DATA por acogerme y brindarme la oportunidad de recorrer este camino. Su apoyo ha sido esencial para mi formación y crecimiento laboral.

A mi director David González por su paciencia, por prestarme apoyo incondicional en cualquier momento y por enseñarme, orientarme y ayudarme siempre. Gracias a él, en gran medida, este proyecto ha salido adelante. Ha sido un placer investigar y desarrollar todo esto juntos, espero que la vida nos traiga muchos otros proyectos con los que seguir creciendo.

A mi directora Vanessa Bueno por ser la impulsora de este proyecto, un café tiene la culpa, pero gracias, gracias por aportar la idea, por tu cercanía, por tenderme la mano cuando más lo he necesitado y por aportar todo tu conocimiento en esto.

A Diego Romero, Jose Ignacio Aznar, David Montal y todos los integrantes de la empresa NTT DATA, junto con las colaboraciones de NTT DATA Brasil, e Innovation Center NTT DATA, que habéis apostado por este proyecto, haciéndolo posible.

A mi madre, una mujer humilde, luchadora, trabajadora y fuerte, la que me ha inculcado los valores que hoy en día me hacen ser quien soy. Quien me ha enseñado a ser feliz, a ser cada día mejor persona y que pase lo que pase siempre se puede ir hacia delante. Ella que siempre está ahí, con su energía imparable, su solución para todo y su confianza plena en mí.

A mi padre, humilde, trabajador y siempre para su familia. A su manera me quiere y a su manera me apoya, pero indudablemente es la manera más maravillosa de hacerme crecer como persona, de ayudarme y de enseñarme que en la vida las cosas cuestan pero que, si es lo que deseas, merece la pena el camino.

A mi hermano Alfonso, sin duda eres un ejemplo a seguir por tu valentía y perseverancia. Estoy segura de que vas a triunfar más de lo que lo haces y de que seguirás ayudando a muchas más personas en tu camino, lo que me hace estar orgullosa de ti y me da fuerzas para seguir creciendo. A mi hermano Ángel, por su bondad y cariño, aunque pequeño y a veces revoltoso me hace estar actualizada de todas las novedades adolescentes.

A mi familia en general y a Miguel por confiar en mí y por apoyarme en este camino y en todos los de mi vida.

RESUMEN

La computación cuántica, una tecnología en auge, promete revolucionar la resolución de problemas en diversos campos e industrias. Aprovechando la potencia de los qubits, los ordenadores cuánticos tienen la capacidad de codificar múltiples valores simultáneamente, superando con creces las limitaciones de la computación clásica.

En el campo de la bioinformática, la computación cuántica es muy prometedora y en este trabajo se ha aplicado para resolver el problema del ensamblaje del genoma *de-novo*. El ensamblaje de genoma es una tarea compleja en genómica, y la computación cuántica puede proporcionar métodos más precisos y eficientes para su resolución. Hemos desarrollado e implementado un algoritmo para resolver el problema del ensamblaje del genoma utilizando técnicas de teoría de grafos y optimización combinatoria. Hemos ejecutado este algoritmo en un simulador de computación cuántica y en un dispositivo de computación cuántica real, y hemos validado su rendimiento en comparación con los enfoques clásicos en términos de velocidad, precisión y escalabilidad en conjuntos de datos sintéticos simulados.

Esta memoria resume nuestros resultados y analiza el potencial, los retos y las limitaciones de la computación cuántica para resolver problemas como éste, que superan rápidamente las capacidades de la computación clásica.

ABSTRACT

Quantum computing, a rapidly advancing technology, holds the promise of revolutionizing problem-solving across various fields and industries. Harnessing the power of qubits, quantum computers have the ability to encode multiple values simultaneously, far surpassing the limitations of classical computing.

In the field of bioinformatics, quantum computing shows great promise, and in this work, it has been applied to tackle the *de-novo* genome assembly problem. Genome assembly is a complex task in genomics, and quantum computing can provide more accurate and efficient methods for its resolution. We have developed and implemented an algorithm to address the genome assembly problem using graph theory and combinatorial optimization techniques. This algorithm has been executed on a quantum computing simulator and a real quantum computing device, and its performance has been validated against classical approaches in terms of speed, accuracy, and scalability on simulated synthetic datasets.

This report summarizes our findings and analyzes the potential, challenges, and limitations of quantum computing in solving such problems that rapidly outpace the capabilities of classical computing.

TABLA DE CONTENIDO

1. INTRODUCCIÓN

El ensamblaje de genoma y la computación cuántica son dos áreas de investigación que han experimentado un rápido desarrollo en las últimas décadas, cada una en su propio campo [1].

El campo de la genómica se remonta a los primeros intentos de secuenciación de ADN en la década de 1970, cuando Frederick Sanger desarrolló el método de secuenciación de Sanger [2], una técnica pionera que permitía la determinación secuencial de fragmentos de ADN. Este enfoque revolucionó la biología molecular y allanó el camino para la era de la secuenciación genómica.

En la década de 1990, se logró un hito importante con el Proyecto del Genoma Humano [3], que buscaba secuenciar y ensamblar el genoma humano completo. Este proyecto masivo sentó las bases para el desarrollo de tecnologías de secuenciación de nueva generación (NGS), que permitieron secuenciar rápidamente genomas enteros a una fracción del costo y el tiempo requeridos anteriormente.

Desde entonces, el campo del ensamblaje de genoma ha avanzado significativamente, con mejoras en las técnicas de secuenciación, algoritmos de ensamblaje y capacidad computacional. Estos avances han permitido el estudio de diversos genomas, incluidos los de organismos no humanos, y han desempeñado un papel fundamental en la comprensión de la estructura y función genómica, así como en la identificación de variantes genéticas relacionadas con enfermedades y rasgos biológicos.

A pesar de los avances realizados en el campo del ensamblaje del genoma, algunos de ellos siguen siendo un gran desafío en bioinformática debido a la complejidad de los datos genómicos y los requerimientos computacionales involucrados. Con la creciente disponibilidad de tecnologías avanzadas, existe un aumento en la demanda de métodos eficientes y precisos para llevar a cabo el ensamblaje del genoma.

En este contexto, se han identificado casos en los que la computación cuántica tiene el potencial de aportar un valor significativo a las industrias de la salud y las ciencias de la vida. En particular, se ha analizado la aplicación de la computación cuántica al problema del ensamblaje del genoma. Se han investigado los resultados obtenidos hasta el momento, así como los desafíos y limitaciones inherentes a la computación cuántica para abordar problemas de esta naturaleza, que rápidamente superan las capacidades de la computación clásica.

La computación cuántica tiene sus raíces en los principios de la física cuántica, una rama de la ciencia que surgió a principios del siglo XX. A medida que se descubrieron y comprendieron mejor los fenómenos cuánticos, los científicos comenzaron a explorar cómo se podrían aplicar estos principios en el campo de la informática.

En la década de 1980, el físico Richard Feynman propuso la idea de la computación cuántica [4], planteando que los sistemas cuánticos podrían realizar cálculos mucho más rápido que los clásicos para ciertos tipos de problemas. Sin embargo, fue en la década de 1990 cuando se lograron los primeros avances significativos en la implementación y manipulación de qubits, las unidades fundamentales de la información cuántica.

En los últimos años, ha habido un creciente interés y desarrollo en la computación cuántica, con varias empresas y laboratorios de investigación que trabajan en la construcción de ordenadores cuánticos de propósito general. Estas máquinas prometen superar las limitaciones de los ordenadores clásicos, lo que podría tener un impacto significativo en campos como la criptografía, la simulación de sistemas complejos y la optimización.

La intersección entre el ensamblaje de genoma y la computación cuántica ha surgido recientemente [5],[6] y [7] como una posible aplicación de la capacidad computacional cuántica para abordar desafíos en el análisis y el ensamblaje de grandes conjuntos de datos genómicos. La computación cuántica podría proporcionar algoritmos y capacidades de procesamiento más eficientes para lidiar con la complejidad de los datos genómicos, acelerando así el ensamblaje y el análisis de genomas completos.

En este proyecto "Ensamblaje de genoma usando técnicas de optimización cuánticas" pretendemos explorar la capacidad y viabilidad del uso de la computación cuántica para el ensamblaje de genoma, mediante la evaluación comparativa de enfoques de computación cuántica y no cuántica. Los objetivos que pretendíamos alcanzar incluían:

- Resolver el problema del ensamblaje de genoma mediante el uso de tecnologías cuánticas.
- Implementar un algoritmo clásico (no cuántico) como referencia. Este enfoque se ha basado en solvers clásicos (como los disponibles en la plataforma de Gurobi). Por otro lado, implementar un algoritmo de computación cuántica para el ensamblaje del genoma, utilizando hardware y software de computación cuántica.
- Comparar los enfoques clásico y cuántico con indicadores clave de rendimiento bien establecidos, como la agilidad, la eficiencia computacional, la precisión y la escalabilidad.
- Averiguar la capacidad de los solvers cuánticos disponibles, es decir, tamaño de problema que es posible resolver a día de hoy.
- Constatar que en biomedicina hay una oportunidad de aplicación de tecnologías cuánticas y de este tipo de enfoques matemáticos y algorítmicos.

El problema se modela en el marco de la Teoría de Grafos y la Teoría de la Optimización Combinatoria, [8] donde, basándose en los solapamientos entre los fragmentos del genoma secuenciado, se debe encontrar el orden correcto de los fragmentos. El enfoque clásico implica el uso de heurísticos, así como de algoritmos exactos para resolver este problema. El enfoque cuántico implica algoritmos heurísticos para expresar el problema como un problema de Optimización Combinatoria y resolverlo en hardware cuántico, que aprovecha las propiedades de la mecánica cuántica para realizar la misma tarea de una manera potencialmente más eficiente.

Este documento resume la definición y el modelado del problema, el conjunto de datos para evaluar el modelo, así como los resultados, la comparación de resultados y las conclusiones. Por último, pero no por ello menos importante, también se han incluido algunas hipótesis de la formulación del modelo, los problemas detectados, correcciones y las lecciones aprendidas.

En conclusión, tanto el ensamblaje de genoma como la computación cuántica han experimentado avances significativos en las últimas décadas. El ensamblaje de genoma ha permitido la secuenciación y reconstrucción de genomas completos, mientras que la computación cuántica ofrece nuevas posibilidades para el procesamiento de datos y la resolución de problemas inmensamente complejos computacionalmente. La convergencia de estos campos podría abrir nuevas oportunidades para mejorar la comprensión genómica y acelerar los avances en la medicina personalizada y la biología. En este trabajo, ofrecemos una visión general de la tecnología cuántica, su desarrollo y su aplicación específica en el campo de la bioinformática.

2. MARCO TEÓRICO

2.1. GENÓMICA

La secuenciación genómica es el proceso de fragmentar una secuencia de ADN para posteriormente ensamblarla y determinar la secuencia completa de ADN de un organismo. Esta tecnología ha experimentado avances significativos en las últimas décadas, permitiendo la secuenciación rápida y precisa de genomas completos. El objetivo principal de la secuenciación genómica es obtener información detallada sobre la estructura, función y variabilidad de los genomas, lo que a su vez proporciona conocimientos fundamentales sobre la biología de los organismos y su relación con la salud humana y animal.

Existen diferentes enfoques y tecnologías para llevar a cabo la secuenciación genómica. La secuenciación de nueva generación (NGS) ha revolucionado el campo, superando las técnicas tradicionales basadas en la secuenciación de Sanger. Existen varias técnicas que usan tecnología NGS, como pueden ser Ilumina, PacBio o Nanopore y permiten la secuenciación masiva y paralela de miles o incluso millones de fragmentos de ADN, lo que resulta en un aumento significativo de la eficiencia y la reducción de los costos.

La secuenciación genómica implica una serie de etapas clave. En primer lugar, se requiere la preparación de la muestra de ADN, que puede ser ADN genómico completo o fragmentos específicos del genoma. Luego, los fragmentos de ADN se amplifican y se preparan en librerías, que son colecciones de fragmentos de ADN individuales listos para la secuenciación. A continuación, se realiza la secuenciación propiamente dicha, donde los fragmentos de ADN se leen y se generan secuencias cortas llamadas "lecturas" o "reads". Una vez que se obtienen las lecturas, el siguiente paso es el ensamblaje del genoma.

El ensamblaje del genoma es el proceso de reconstruir el genoma completo a partir de las lecturas secuenciadas. Dado que las lecturas generadas por la secuenciación son fragmentos cortos, es necesario superponer y unir estas lecturas para construir secuencias más largas y contiguas que representan el genoma original.

El ensamblaje de genoma puede ser un desafío debido a varias características genómicas, como la repetitividad y la presencia de secuencias únicas. Los algoritmos de ensamblaje utilizan diferentes estrategias para superponer y unir las lecturas secuenciadas, con el objetivo de reconstruir el orden correcto y la orientación de los fragmentos del genoma.

El resultado final del ensamblaje es un conjunto de secuencias llamadas "contigs" que representan regiones contiguas del genoma. Los contigs se alinean y se ordenan en función de información adicional, como mapas genéticos o datos de secuenciación de largos alcances, para generar una secuencia genómica final más completa y precisa. El resultado del ensamblaje de genoma puede presentar varios escenarios, que los contigs tengan la misma largura que los cromosomas (por lo que tendríamos un ensamblaje completo) o podrían tener varios contigs que representan un mismo cromosoma y no seamos capaces de unirlos.

2.1.1 ENSAMBLAJE DE NOVO

El ensamblaje 'de novo', también conocido como ensamblaje sin referencia, es una técnica utilizada en bioinformática y genómica para reconstruir la secuencia completa de un genoma a partir de datos de secuenciación sin una referencia previa de otro genoma relacionado. Como hemos mencionado, partiendo de las secuencias fragmentadas del genoma llamadas

reads, el problema consiste en encontrar el orden de las reads que hace que se reconstruya el genoma. A diferencia del ensamblaje basado en referencia, donde se utiliza un genoma de referencia como guía, el ensamblaje 'de novo' se realiza cuando no se dispone de una secuencia de referencia cercana o cuando se está investigando un organismo completamente nuevo.

El ensamblaje 'de novo' es un proceso complejo que implica la superposición y alineación de fragmentos secuenciados para identificar regiones de similitud y construir una secuencia genómica contigua y precisa. El objetivo principal es reconstruir el orden y la orientación correctos de los fragmentos de ADN para obtener una representación coherente del genoma original.

El proceso de ensamblaje 'de novo' generalmente se divide en varias etapas. Primero, se generan las lecturas de secuenciación a partir de los fragmentos de ADN. Luego, estas lecturas se someten a un proceso de preprocesamiento, que puede incluir filtrado de calidad, eliminación de adaptadores y corrección de errores de secuenciación. Posteriormente, se realiza la superposición y alineación de las lecturas para identificar regiones de similitud.

Existen diferentes algoritmos y enfoques para realizar el ensamblaje 'de novo', que varían en complejidad y eficiencia. Algunos algoritmos populares incluyen el ensamblaje de overlap-layout-consensus (OLC), el ensamblaje de grafo de De Bruijn y el ensamblaje basado en cadenas. Cada uno de estos métodos tiene sus propias ventajas y desafíos, y la elección del algoritmo depende del tipo de datos de secuenciación y de las características del genoma objetivo.

Es importante tener en cuenta que el ensamblaje 'de novo' puede ser especialmente desafiante en genomas grandes, complejos o altamente repetitivos. En estos casos, se requiere el uso de estrategias adicionales, como la secuenciación de largos alcances o la integración de datos de múltiples plataformas de secuenciación, para obtener un ensamblaje más completo y preciso.

En resumen, el ensamblaje 'de novo' es una técnica fundamental en genómica y bioinformática que permite reconstruir la secuencia completa de un genoma sin una referencia previa. A través de la superposición y alineación de lecturas secuenciadas, se busca obtener una representación coherente y precisa del genoma original. Esta metodología es crucial para investigar organismos nuevos, identificar variantes genéticas y comprender la diversidad genómica en diferentes especies.

2.1.2 APLICACIONES

Un genoma contiene toda la información genética que hace único a un organismo. Ensamblar el genoma a partir de los fragmentos o lecturas producidos por las técnicas de secuenciación permite estudiar esta información en detalle.

Una de las principales aplicaciones del ensamblaje del genoma es la investigación médica. Al secuenciar y ensamblar los genomas de los individuos, los científicos pueden identificar variaciones genéticas que pueden estar asociadas a determinadas enfermedades o afecciones. Esta información puede utilizarse para desarrollar nuevos tratamientos o terapias dirigidos a las causas genéticas subyacentes de las enfermedades o para el estudio de nuevas especies y variantes o incluso en el ámbito de la microbiota humana, un tema cada vez más estudiado donde la caracterización de los microbiomas es compleja e importante.

El ensamblaje de genomas también desempeña un papel crucial en campos como la biología evolutiva y la ecología. Comparando los genomas de distintas especies, los investigadores pueden conocer las relaciones evolutivas entre ellas y comprender cómo se han adaptado los distintos organismos a su entorno.

El ensamblaje de genomas también tiene importantes aplicaciones en los campos de la biotecnología y la biología sintética. Esto es especialmente relevante en el campo de la metagenómica, donde el ensamblaje de genomas para su comparación puede aportar conocimientos sobre la función génica de organismos específicos que podrían tener aplicaciones biotecnológicas. Al secuenciar y ensamblar con precisión los genomas de los organismos, los científicos pueden comprender mejor la base genética de los rasgos y utilizar este conocimiento para desarrollar nuevos productos o tecnologías. Por ejemplo, las técnicas de ingeniería genética pueden utilizarse para modificar el código genético de un organismo, lo que permite a los investigadores crear nuevas cepas con características deseables, como una mayor resistencia a las enfermedades o un mejor rendimiento de un determinado producto. Además, el ensamblaje del genoma puede ayudar a diseñar genomas sintéticos de organismos como bacterias capaces de producir biocombustibles, fármacos u otros bioproductos.

El ensamblaje de genomas se aplica en una amplia gama de campos, desde la investigación básica en genómica pasando por la medicina personalizada y la biotecnología hasta la agricultura. En conjunto, es una herramienta fundamental para el avance del conocimiento, la comprensión de la genética, la biología, la innovación en los sectores de las ciencias de la vida y la salud.

2.2 COMPUTACIÓN CUÁNTICA

La computación cuántica ha experimentado un auge significativo en la última década, permitiendo empezar a abordar problemas del mundo real que antes eran imposibles o poco prácticos de resolver utilizando algoritmos clásicos. Es una tecnología en desarrollo que tiene el potencial de transformar campos como las finanzas, la criptografía, la logística y la industria farmacéutica, entre otros.

La computación cuántica pretende resolver problemas informáticos utilizando sistemas físicos que se rigen por las leyes de la Física Cuántica donde los bits se sustituyen por bits cuánticos o qubits. Actualmente se están probando distintos tipos de qubits: superconductores, átomos neutros, iones atrapados, basados en la fotónica, etc. En todos los casos, surgen en ellos los principios de la mecánica cuántica de: superposición, interferencia y entrelazamiento, lo que los hace capaces de codificar exponencialmente muchos más valores, de forma simultánea, que sus homólogos clásicos.

- **Superposición**: Propiedad cuántica donde los qubits pueden existir en uno o múltiples estados a la vez.
- **Interferencia**: Los estados individuales de los qubits se combinan colectivamente como si fueran ondas y pueden producir interferencias, constructivas o destructivas. Los algoritmos cuánticos buscan reforzar por interferencia los estados que son solución del problema y anular los que no lo son.
- **Entrelazamiento**: Propiedad cuántica en la que dos o más partículas están altamente correlacionadas, de modo que, conociendo el estado de uno, se conoce inmediatamente el estado del otro.

En este contexto, surgen los ordenadores cuánticos, dispositivos que explotan los principios fundamentales de la mecánica cuántica para transformar la forma en la que concebimos la computación.

El principio de superposición cuántica aplicada a estos ordenadores cuánticos desafía las restricciones binarias de los bits clásicos, es decir, en vez de ser 0 o 1, pueden estar en múltiples estados a la vez.

Por otro lado, la interferencia cuántica es un hito crucial en ordenadores cuánticos para el abordaje de problemas que se consideraban intratables en ordenadores clásicos debido a encaminar a los estados que son solución del problema.

Por último, el entrelazamiento cuántico conecta los qubits permitiendo un procesamiento paralelo masivo que implica velocidad de cálculo inimaginable.

La velocidad exponencial con la que los ordenadores cuánticos pueden abordar tareas complejas augura un futuro prometedor para resolver ciertos desafíos tecnológicos.

Actualmente existen dos enfoques principales en la fabricación de ordenadores cuánticos:

- **Basado en puertas lógicas cuánticas**: Se basan en el uso de qubits que se manipulan mediante puertas cuánticas, que son análogas a las puertas lógicas clásicas, pero operan en dominio cuántico. Al combinar estas puertas, se pueden realizar cálculos y operaciones complejas en los qubits para realizar cálculos cuánticos. Se han logrado importantes avances teóricos en el desarrollo de algoritmos cuánticos para este tipo de ordenadores, como el algoritmo de Shor para factorización y el algoritmo de Grover para búsqueda. Sin embargo, la implementación práctica de estos sistemas enfrenta desafíos significativos debido a la sensibilidad de los qubits, a los errores cuánticos y al ruido.

- **Basado en el Principio Adiabático:** Se trata de hacer evolucionar un sistema cuántico de manera adiabática hacia un estado controlado por un Hamiltoniano (operador que describe la energía total del sistema de qubits) que codifica un problema a resolver. El estado de menor energía representa la solución óptima. El Principio Adiabático garantiza que, bajo ciertas condiciones, se puede empezar con un Hamiltoniano diferente, donde el estado fundamental es fácil de preparar, y hacerlo evolucionar adiabáticamente, hasta convertirlo en el Hamiltoniano del problema. Entonces el sistema de qubits habrá permanecido en el estado fundamental para ese Hamiltoniano final, y representa la solución óptima al problema planteado. Una aproximación práctica a ese planteamiento teórico es el Quantum Annealing. Estos ordenadores actualmente son útiles para resolver problemas de optimización y simulación cuántica.

La computación cuántica está aún en sus primeras fases de desarrollo, y actualmente existen varias limitaciones, en particular de hardware, ya que la tecnología para construir ordenadores cuánticos está aún en sus primeras fases de desarrollo, y los sistemas actuales de qubits cometen muchos errores, lo que requiere complejas técnicas de corrección de errores para mantenerse estables. Otro problema es la escalabilidad: no es seguro que la computación cuántica pueda ampliarse para resolver problemas en corto o medio plazo, ya que el número de qubits necesarios para resolver problemas complejos aumenta exponencialmente.

2.2.1 SINERGIA COMPUTACIÓN CUÁNTICA Y GENÓMICA

La computación cuántica tiene el potencial de acelerar la resolución de problemas complejos. Es el caso de la Genómica, donde actualmente existen problemas de optimización del genoma [5], [6] y [7], cuya complejidad computacional aumenta rápidamente. En ese caso, los enfoques de computación clásica encuentran grandes dificultades al intentar resolverlos, ya que los métodos informáticos tradicionales se basan en algoritmos de fuerza bruta para resolver problemas matemáticos complejos, lo que puede resultar caro y llevar mucho tiempo. En cambio, los ordenadores cuánticos pueden realizar cálculos complejos de forma mucho más rápida y eficiente, lo que ayuda a los investigadores a procesar mayores cantidades de datos genómicos en menos tiempo. Además, puede conducir a ensamblajes genómicos más precisos, proporcionar nuevos algoritmos y métodos computacionales que permiten obtener diagnósticos más rápidos de enfermedades genéticas.

La computación cuántica también puede contribuir al ensamblaje de genomas, permitiendo a los investigadores explorar nuevos enfoques de secuenciación y análisis genéticos. Por ejemplo, los algoritmos cuánticos podrían utilizarse para analizar los patrones y características únicas de los datos genéticos, permitiendo a los investigadores identificar genes específicos o variaciones genéticas con mayor precisión y exactitud.

Los algoritmos de optimización cuántica son especialmente relevantes para el sector sanitario y pueden ofrecer una ventaja competitiva cuando se trata de problemas computacionales complejos, así como de aquellos que requieren una gran capacidad de cálculo.

2.2.2 SINERGIA COMPUTACIÓN CUÁNTICA Y SALUD

En sanidad, la computación cuántica también puede utilizarse para la obtención de imágenes médicas y la medicina personalizada. Los algoritmos cuánticos podrían analizar grandes conjuntos de imágenes médicas, lo que permitiría a los médicos detectar anomalías y diagnosticar enfermedades con mayor rapidez. La medicina personalizada implica adaptar los tratamientos a la composición genética específica de un individuo, y la computación cuántica puede ayudar a analizar grandes conjuntos de datos de información genética para identificar los tratamientos óptimos. Además, la analítica sanitaria podría ser más rápida y precisa en el análisis de conjuntos de datos sanitarios a gran escala mediante el uso de tecnologías basadas en la computación cuántica. Por ejemplo, podría acelerar el análisis de las historias clínicas electrónicas (HCE) y permitir a los profesionales de la salud y la asistencia sanitaria realizar análisis más precisos al tiempo que se identifican patrones y tendencias en los brotes de enfermedades y se contribuye a tomar decisiones mejor informadas sobre, por ejemplo, la optimización y eficacia de las prescripciones.

En el ámbito de las ciencias de la vida, también se han identificado impactos concretos en los que la computación cuántica podría marcar la diferencia. Es el caso del descubrimiento de fármacos: La computación cuántica podría utilizarse para simular el comportamiento de las moléculas y predecir cómo interactuarían con otras moléculas, lo que podría acelerar el proceso de descubrimiento de fármacos [9]. Esto podría conducir al desarrollo de nuevos tratamientos para enfermedades que actualmente son difíciles de tratar. La medicina de precisión [10] también es una disciplina que podría beneficiarse de la computación cuántica, ya que podría utilizarse para analizar grandes cantidades de datos genómicos y clínicos con el fin de desarrollar planes de tratamiento personalizados para los pacientes. Esto podría

mejorar la eficacia de los tratamientos y reducir el riesgo de efectos secundarios adversos. En cuanto al plegamiento de proteínas, la computación cuántica tiene el potencial de utilizarse para simular el plegamiento de proteínas, que es un proceso crítico para entender cómo funcionan las proteínas y cómo pueden ser objeto de fármacos [11], [12]. Esto podría conducir al desarrollo de nuevos tratamientos para enfermedades como el cáncer y el Alzheimer.

En general, la computación cuántica tiene el potencial de acelerar los descubrimientos científicos y mejorar nuestra comprensión de los sistemas biológicos complejos, dando lugar a nuevos tratamientos para enfermedades y mejores resultados sanitarios para los pacientes.

Representa un cambio de paradigma revolucionario en el campo de las ciencias de la vida y otras ciencias y ofrece a investigadores y profesionales oportunidades sin precedentes para acelerar los descubrimientos científicos, mejorar nuestra comprensión de los sistemas biológicos complejos y desvelar los misterios de la biología y la bioinformática. A medida que la tecnología madura y la inversión en investigación cuántica sigue acelerándose, el potencial de la computación cuántica para revolucionar la asistencia sanitaria e impulsar la innovación médica es realmente ilimitado. Aunque aún quedan retos por superar, la computación cuántica desempeñará un papel crucial en la configuración del futuro de la sanidad y en la transformación de nuestra forma de pensar sobre el funcionamiento fundamental de la vida misma.

3. METODOLOGÍA

En este trabajo, hemos hecho uso de dos herramientas para abordar este problema de Ensamblaje de Genoma y comparar tanto el enfoque cuántico como el clásico.

Por un lado, se han utilizado servicios de D-Wave Systems, que es una empresa que proporciona acceso a ordenadores cuánticos (del tipo Quantum Annealing), APIs, librerías de software y herramientas de desarrollo relacionadas. En particular, se ha hecho uso de la plataforma D-Wave Leap que permitía tiempo de acceso a distintos solvers.

Por otro lado, se ha hecho uso de Gurobi Optimizer que es una plataforma con diversos solvers clásicos para formular y resolver problemas de optimización.

Ejecutaremos cuatro solvers: El Hybrid Solver, el Advantage 6.1, el Simulated Annealing Solver y Gurobi.

- **Solver Simulated Annealing *(D-Wave Simulated Annealing)*.** Es un algoritmo de optimización que se ejecuta en hardware clásico y trata de encontrar soluciones a problemas de minimización de costes. Es muy similar a un algoritmo de “ascenso a la colina” o *Hill Climbing* para buscar óptimos globales en una hipersuperficie definida por una función de coste a optimizar. Es uno de los algoritmos que más se utilizan cuando se comparan las prestaciones de los enfoques cuánticos y de inspiración cuántica.

  El algoritmo es en muchos aspectos similar al quantum annealing (pero no es un simulador de quantum annealing). Con la diferencia de que el simulated annealing utiliza un parámetro similar a la temperatura que controla la capacidad de saltar fuera de las soluciones de mínimos locales encontradas durante la búsqueda y, finalmente, alcanzar un mínimo global (es decir, la solución óptima). En los dispositivos de quantum annealing reales, esto se rige por el efecto de túnel cuántico que surge en los sistemas físicos a escalas cuánticas.

El algoritmo comienza con un candidato a solución aleatorio y, en cada iteración, genera un nuevo candidato a solución que puede ser o no mejor que el anterior, eligiendo un criterio basado en la diferencia entre la función de coste de la nueva solución y la función de coste de la solución actual, así como esa temperatura ficticia, que disminuye con el tiempo.

El algoritmo sigue iterando hasta que la temperatura ficticia alcanza un valor mínimo. En ese momento, el algoritmo se detiene y devuelve la solución actual como la mejor encontrada.

- **Solver Advantage 6.1 QPU *(D-Wave Quantum Advantage 6.1)***. El solver Advantage es un solucionador cuántico puro que utiliza los procesadores qubit de D-Wave (o QPU) para resolver problemas de optimización. Estos procesadores implementan el ya mencionado proceso conocido como "Quantum annealing" para buscar la solución óptima.

  La serie Advantage de ordenadores cuánticos de D-Wave (Advantage6.1 es su última versión) contiene más de 5000 qubits y 15 acopladores por qubit, es decir, más de 35000 acopladores en total. El número de qubits determina el número de variables del problema que pueden asignarse al procesador, y el número de acopladores determina cuántos coeficientes que relacionan pares de variables pueden representarse en la QPU.

  Sin embargo, debido a la conectividad limitada que proporciona este número de acopladores, a menudo es necesario combinar varios qubits en grupos que actúen como qubits lógicos únicos para igualar el número de coeficientes cuadráticos del problema. Esto reduce el número de qubits disponibles para las variables y limita el tamaño de los problemas que pueden resolverse directamente en una sola QPU.

  El quantum annealing funciona en estos procesadores aplicando sesgos y acoplamientos dependientes del tiempo a los qubits, durante una duración aproximada de 1-200 microsegundos (20 microsegundos es el valor predeterminado), asemejándose a un proceso de annealing (o enfriamiento), pero de naturaleza cuántica. En las etapas finales, el Efecto Túnel Cuántico ayuda a escapar de las soluciones de estados mínimos locales. Al final de este ciclo, los sesgos y valores de acoplamiento corresponden a los coeficientes del problema a resolver, y los valores de los qubits codifican los valores de solución para las variables del problema.

  Es necesario ejecutar muchas muestras o disparos de un problema para recopilar suficientes estadísticas y garantizar que se encuentren soluciones óptimas o, al menos, suficientemente buenas. D-Wave impone algunos límites a este número.

- **Solver híbrido cuántico-clásico *(D-Wave Hybrid Solver)***. El solver híbrido combina la potencia de la computación clásica y cuántica para resolver problemas de optimización. Este enfoque permite resolver muchos problemas que no caben en las actuales QPU Advantage, debido a su tamaño (número de variables y coeficientes). Puede manejar problemas con miles, hasta un millón, de variables.

  Este solver utiliza flujos de trabajo cuántico-clásicos: computación clásica, para preparar una solución inicial, y computación cuántica, para refinar y mejorar la solución. Hay que establecer un límite de tiempo o utilizar un valor por defecto calculado por el solver que depende del tamaño del problema.

Los flujos de trabajo combinan un módulo heurístico clásico que explora el espacio de soluciones, y un módulo cuántico que intenta resolver partes del problema en la QPU, guiado por el módulo heurístico hacia áreas prometedoras del espacio de soluciones o hacia soluciones mejoradas. Una vez transcurrido el tiempo límite, se devuelve la mejor solución encontrada por la heurística.

- **Solver clásico con Gurobi *(Gurobi Classical Linear)*** Gurobi es un optimizador de última generación para resolver problemas de programación matemática. Está demostrado que el solver puede proporcionar soluciones óptimas para problemas de optimización lineal (LP), cuadrática (QP) y mixta entera (MIP). Debido a su flexibilidad, es posible evaluar diferentes enfoques para el mismo problema con el mismo optimizador. Para resolver numéricamente problemas de programación lineal, Gurobi utiliza el método Simplex, evaluando en cada iteración si se satisfacen todas las restricciones y si la función de coste es mínima, con una tolerancia de 10^{-4}.

4. DESCRIPCIÓN DEL PROBLEMA

Este trabajo pretende abordar uno de los mayores retos de la genómica actual: el ensamblaje asistido por ordenador de secuencias genómicas fragmentadas, las llamadas reads o lecturas.

El problema del ensamblaje del genoma puede formularse como un problema de optimización combinatoria en un grafo (Figura 1b), en el que los nodos representan las lecturas (Figura 1a) y las conexiones entre pares de nodos dependen de los solapamientos mutuos entre las lecturas.

Figura 1. a) Ejemplo de secuencia genómica con longitud 16 pares de bases (bp). b) Grafo dirigido que conecta las parejas de nodos con solapamiento.

Existen múltiples formas de resolver este tipo de problemas de optimización. En este caso concreto, nos basamos en el Problema del Viajante [13] (Traveling Salesperson Problem, TSP), en el que el camino de distancia total mínima que conecta todos los nodos (sujeto a algunas restricciones) proporciona el orden correcto de las reads, permitiendo así ensamblar el genoma original (Figura 2).

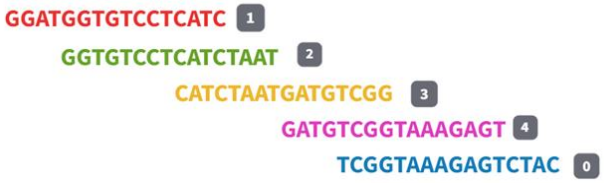


Figura 2. Secuencia del genoma del ejemplo donde se muestran los solapamientos entre las reads.

Este tipo de distancia entre dos lecturas puede calcularse como la longitud total de una lectura menos el grado de solapamiento entre ellas (asumiendo que las lecturas tienen siempre la misma longitud). Es decir, cuanto mayor sea el solapamiento, menor será la distancia. Es importante señalar que la distancia del nodo 1 al nodo 2 no es igual, en general, a la distancia del nodo 2 al 1, lo que implica que estamos trabajando con un grafo completo (potencialmente todos pares de nodos están conectados) y dirigido (en cada par de nodos conectados, es relevante cuál es origen y cuál destino).

Formulamos matemáticamente el problema anterior como un problema de optimización binaria cuadrática sin restricciones, o QUBO, que se representa en una matriz de coeficientes lineales y cuadráticos que se puede enviar a un ordenador cuántico para que la resuelva.

Para comparar el rendimiento del modelo cuántico, se evaluaron los modelos clásicos de ensamblaje de genomas desarrollados. Es posible resolver el TSP con enfoques lineales o no lineales y desacoplando las restricciones de la función de coste.

En este estudio, el modelo clásico desarrollado fue el enfoque lineal debido a su menor complejidad y a su rendimiento típicamente superior en comparación con el no lineal ya que sabemos que es lo mejor que se puede esperar de los algoritmos clásicos.

Los modelos cuánticos que hemos llevado a cabo son dos: Formulación lineal y no lineal (cuadrático). En los solvers cuánticos de los que vamos a hacer uso es necesario introducir un QUBO para obtener una solución por lo que vamos a definir uno para cada una de las dos formulaciones.

Por tanto, usaremos las soluciones clásicas obtenidas con Gurobi como referencia y sobre ellas compararemos las soluciones cuánticas de las formulaciones lineal y cuadrática.

Esta comparación nos permitirá conocer cuál de las dos formulaciones cuánticas es más efectiva frente a la clásica lineal.

4.1 FORMULACIÓN CUADRÁTICA

Esta formulación está centrada en los nodos y los pasos temporales lo que implica una función de coste cuadrática donde se multiplican dos variables binarias (x), una para cada nodo y otra para cada paso temporal. En un problema de n nodos tendremos n^2 variables ya que todos los nodos están conectados con todos (grafo completo dirigido).

Para definir el QUBO escribimos la función de coste a minimizar (1) y las restricciones que deben cumplir las soluciones (2) y (3).

- Función de coste (1): Minimizar la suma de las distancias a lo largo del camino del grafo.

$$minimize: \sum_{i=0}^{n-1}\sum_{j=0}^{n-1} w_{ij} \sum_{p=0}^{n-1} x_{i,p}x_{j,p+1} \qquad (1)$$

donde ω_{ij} es la distancia entre las lecturas i y j, n es el número de lecturas, y x_{ip} y $x_{j,p+1}$ son valores binarios donde el producto $x_{ip} \cdot x_{j,p+1}$ será 1 si el nodo i está en el tiempo p y el nodo j está en el tiempo siguiente p+1, y será 0 en cualquier otro caso.

- Restricción (2): Cada nodo está incluido en el camino una única vez.

$$\forall i \in \{0...(n-1)\}, \sum_{p=0}^{n-1} x_{i,p} = 1 \tag{2}$$

- Restricción (3): Cada paso en el camino contiene sólo un nodo.

$$\forall p \in \{0...(n-1)\}, \sum_{i=0}^{n-1} x_{i,p} = 1 \tag{3}$$

Las restricciones implican que en las soluciones no puede aparecer el mismo nodo dos veces ni que aparezcan dos nodos en un mismo paso del camino.

Para formar el QUBO, dada la función de coste se acoplan las restricciones cuadratizadas con un multiplicador de Lagrange. Una vez definido el QUBO, la expresión a minimizar viene dada por (4):

$$minimize : y = x^T Q x \tag{4}$$

4.2 FORMULACIÓN LINEAL

En este caso, la formulación lineal está basada en las conexiones, en vez de tener nodos y pasos temporales, nuestras variables serán las conexiones. Esto implica que no tenemos todos los nodos conectados con todos y el tamaño del problema será el número de conexiones entre nodos. Es decir, para un problema de n nodos, habrá m variables, correspondientes a las m conexiones con solapamiento entre nodos.

Para definir el QUBO escribimos la función de coste a minimizar (5) y las restricciones que deben cumplir las soluciones (6) y (7).

- Función de coste (5): Minimizar la suma de las distancias a lo largo del camino del grafo.

$$minimize : \sum_{i=0}^{n-1} \sum_{j \neq i, j=0}^{n-1} w_{ij} x_{i,j} \tag{5}$$

donde ω_{ij} es la distancia entre las lecturas i y j, n es el número de lecturas, y x_{ij} es una variable binaria que tendrá valor 1 si i es seguido por j en la trayectoria hamiltoniana y 0 si no lo es. Cabe señalar que x es una matriz de n por n, donde la línea i corresponde al origen y j es el destino para la trayectoria dada. Las restricciones de este problema deben garantizar que, para cada nodo, hay una arista que va a otro nodo y otra que llega desde otro nodo, así las dos restricciones de esta formulación serán:

- Restricción (6): Para cada nodo de destino, sólo haya un nodo de origen.

$$\forall j \in \{0...(n-1)\}, \sum_{j \neq i, i=0}^{n-1} x_{i,j} = 1 \quad (6)$$

- Restricción (7): Para cada nodo de origen, sólo haya un nodo de destino.

$$\forall i \in \{0...(n-1)\}, \sum_{j \neq i, j=0}^{n-1} x_{i,j} = 1 \quad (7)$$

Para formar el QUBO, dada la función de coste se acoplan las restricciones cuadratizadas con un multiplicador de Lagrange. Una vez definido el QUBO, la expresión a minimizar viene dada por (8):

$$minimize : y = x^T Q x \quad (8)$$

4.3 RESOLUCIÓN

El objetivo es encontrar una combinación de variables binarias (x) que minimice la energía de la matriz QUBO (Q), lo que corresponde a encontrar el orden óptimo cerrado de la secuencia de lecturas.

Cuando hablamos de camino cerrado o abierto hacemos referencia a genomas circulares o lineales respectivamente.

Genomas circulares son estructuras de moléculas de ADN que se encuentran en muchos organismos, especialmente bacterias y algunos virus. En un genoma circular, los extremos de la molécula están unidos formando un círculo cerrado. Esto significa que no hay extremos distintos como en un genoma lineal.

Los genomas lineales son estructuras de moléculas de ADN donde los extremos están libres y no están conectados formando un círculo cerrado.

Los genomas circulares carecen de extremos y los lineales tienen dos extremos distintos. A la hora de replicar el ADN, un genoma circular permite el comienzo en cualquier punto específico y continuar en ambas direcciones a diferencia de un genoma lineal que es unidireccional.

El algoritmo que hemos desarrollado es capaz de resolver ambos tipos de genoma. Para el caso de genoma circular la solución cierra el camino completo y para el caso de genoma lineal dada la solución podemos conocer dónde empieza y acaba nuestro genoma (Figura 3a).

Dado el modelo y la formulación de nuestro problema, lo lanzamos a los diferentes solvers para resolver el algoritmo y obtenemos soluciones al ensamblaje del genoma que nos permite finalmente obtener el camino solución (Figura 3a) del ensamblaje del genoma (Figura 3b).

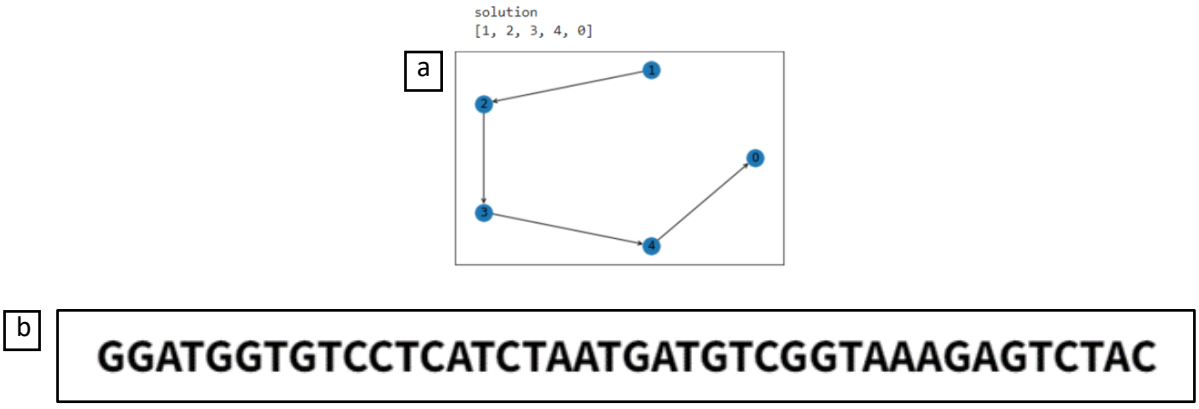


Figura 3. a) Camino óptimo en el grafo dirigido que representa el orden correcto de las lecturas. b) El genoma ensamblado, con las lecturas superpuestas en el orden correcto.

5. GENOMAS

Las tecnologías de secuenciación actuales producen lecturas que son los fragmentos del genoma y suelen ser bastante cortas (100-200 bp), lo que dificulta su ensamblaje. Algunos secuenciadores recientes pueden producir lecturas más largas (10.000 bp), pero suelen contener más errores de secuenciación.

Dado que es imposible distinguir un error de un nucleótido real, se necesita una gran cantidad de datos (cobertura). Esto implica que los algoritmos de ensamblaje tienen que tratar con un elevado número de lecturas para ensamblar un genoma, lo que suele llevar mucho tiempo.

Actualmente, en la mayoría de los secuenciadores reales se generan muchas lecturas para asegurar esa cobertura completa del genoma, predominando las lecturas de longitud corta. Sin embargo, para nuestro enfoque, esto suponía un problema, ya que el tamaño del problema (número de variables) escala con n^2, donde n es el número de nodos, es decir, de lecturas. Las variables n^2 escalan muy rápidamente, y esto supone un problema para los ordenadores cuánticos disponibles en la actualidad. Para resolver este problema, hemos generado nuestros propios datos simulados. De este modo, podemos elegir el número de lecturas y su longitud, lo que facilita el control de la capacidad de nuestro algoritmo y la obtención de resultados precisos.

Una vez definida la capacidad y los parámetros que regulan el modelo, generamos una base de datos con diferentes escenarios que proviene de un simulador de secuenciación que hemos creado para obtener lecturas de genomas que nuestro algoritmo pueda resolver.

Una vez probado el modelo y disponible la versión final, procederemos a resolver los escenarios de datos sintéticos que expondremos más adelante. El objetivo es encontrar una solución óptima para cada uno de ellos, que satisfaga las restricciones impuestas. Mostraremos estos resultados, al tiempo que compararemos las soluciones obtenidas mediante el enfoque cuántico con D-Wave con el enfoque clásico.

Dado este conjunto de soluciones obtenidas a partir de los distintos solvers y enfoques, hay varios parámetros e indicadores que pueden compararse. Nos centramos en estos dos aspectos principales:

- **Energía de las soluciones**. La energía mínima que hemos sido capaces de encontrar con cada uno de los solvers y para cada escenario del problema.

- **Tiempo de computación.** Tiempo requerido por cada solver en cada escenario. Este parámetro tiene algunas peculiaridades que afectan a la comparación y que explicaremos más adelante.

La energía óptima para cada escenario es un parámetro que conocemos y hemos podido calcular debido a la generación de la base de datos sintética. Obtenidas las soluciones de cada solver vamos a clasificarlas en función de su energía y de si cumplen las restricciones del problema o no:

- **Soluciones óptimas**, tienen una energía correspondiente a la energía óptima para el escenario del problema específico.
- **Soluciones factibles**, que no alcanzan la energía óptima, pero satisfacen todas las restricciones.
- **Soluciones inviables**, que no cumplen algunas de las restricciones, aunque su energía podría ser inferior a la energía óptima (las descartamos para nuestro análisis).

Alcanzar la energía óptima con al menos una solución es importante para demostrar que un determinado enfoque puede encontrar las mejores soluciones posibles, dadas las restricciones y los parámetros.

Ahora, mostraremos las bases de datos generadas donde, en un primer momento, trabajamos con el bacteriófago phiX174 que tiene 5386 pares de bases [14] y al analizar los resultados que también veremos a continuación, pudimos aumentar el tamaño del genoma y trabajar con uno más grande, el SARS-CoV-2 con ~30000 pares de bases (29903 en concreto) [15].

5.1 BACTERIÓFAGO

5.1.1 BASE DE DATOS

Para generar las primeras secuencias de nuestro estudio, partimos del genoma del bacteriófago phiX174, que tiene 5386 pares de bases. Introdujimos varios parámetros ajustables para crear nuestro conjunto de datos. La tabla 1 muestra los escenarios de datos generados con los siguientes parámetros.

- En primer lugar, la longitud total del genoma (Longitud), como se ha mencionado anteriormente es el bacteriófago phiX174.
- En segundo lugar, variamos el número de lecturas o nodos (Num. Reads), de 5 a 30 en intervalos de 5 nodos.
- También ajustamos la longitud de las lecturas (Longitud reads) para que coincidiera con el número de nodos necesarios para garantizar que se solapaban lo suficiente como para cubrir la mayor parte del genoma (sólo por comodidad; esto no tiene ningún efecto en nuestros enfoques). Cabe señalar que en todos los casos hemos cubierto todo el genoma, longitud cubierta = 5386.
- Hemos probado diferentes rangos de solapamiento entre las lecturas, incluyendo solapamiento bajo, medio y alto, para todos los casos de nodos. La tabla muestra el rango de solapamientos en porcentaje entre el mínimo (Min.Solap %) y el máximo (Max.Solap %).
- Por último, la energía óptima se refiere a la cantidad mínima de energía (o valor óptimo de la función de coste) que debe obtenerse al resolver el problema, como referencia. Corresponde a la mejor solución esperada de la ruta del grafo para ensamblar el genoma.

Longitud	Num. Reads	Longitud reads	Min. solap %	Max. solap %	Energía
5386	05	1269	15	25	0.98337
5386	10	0652	15	25	0.88667
5386	15	0449	15	25	0.83911
5386	20	0334	15	25	0.83576
5386	25	0270	15	25	0.82185
5386	30	0221	15	25	0.83263
5386	05	1790	45	55	0.73258
5386	10	0985	45	55	0.60081
5386	15	0671	45	55	0.56929
5386	20	0524	45	55	0.53868
5386	25	0412	45	55	0.54270
5386	30	0348	45	55	0.53237
5386	05	2974	75	85	0.51956
5386	10	1857	75	85	0.34968
5386	15	1466	75	85	0.28122
5386	20	1097	75	85	0.27021
5386	25	0933	75	85	0.24966
5386	30	0767	75	85	0.24937
5386	05	1732	20	80	0.75004
5386	10	0841	20	80	0.69980
5386	15	0645	20	80	0.59285
5386	20	0529	20	80	0.53510
5386	25	0441	20	80	0.50884
5386	30	0332	20	80	0.55919
5386	05	1753	40	60	0.74578
5386	10	0973	40	60	0.60817
5386	15	0681	40	60	0.56154
5386	20	0498	40	60	0.56606
5386	25	0407	40	60	0.54932
5386	30	0343	40	60	0.54014

Tabla 1. Base de datos generada bacteriófago phiX174.

Al controlar estos parámetros, podemos crear un conjunto de datos adaptado a nuestras necesidades específicas, lo que nos permite probar la escalabilidad y precisión de nuestro algoritmo en diferentes condiciones.

El grado de solapamiento induce un grado variable de complejidad en el problema, que se añade a la complejidad inducida por la escala (número creciente de nodos o lecturas). Cuanto menores sean los solapamientos, más similares serán las distancias modeladas entre pares de lecturas, y más difícil será encontrar un camino más corto óptimo que sitúe las lecturas en el orden correcto.

Hemos querido dibujar algunos escenarios que parecen realistas o alcanzables, y otros que son casos menos reales, pero que representan un reto interesante para nuestros enfoques. El significado de cada rango de solapamiento es el siguiente:

- 15-25 %: Solapamientos medios muy pequeños, muy concentrados en torno a la media.
- 45-55 %: Solapamientos medios moderados, muy concentrados en torno a la media.
- 40-60 %: Solapamientos medios moderados, con cierta dispersión en torno a la media.
- 20-80 %: Solapamientos medios moderados, con mucha más dispersión en torno a la media, con fracciones importantes de solapamientos muy pequeños y muy altos.
- 75-85 %: Solapamientos medios muy grandes, muy concentrados en torno a la media.

Por un lado, los rangos de solapamiento que inducen una mayor complejidad, a priori, son los rangos 15-25 y 20-80. Por otro lado, los rangos de solapamiento más realistas según las técnicas y prácticas actuales de secuenciación son los rangos 45-55, 40-60 y 75-85, que también son a priori menos complejos.

5.1.2 RESULTADOS

Los solvers utilizados fueron los descritos anteriormente: Gurobi Linear solver, D-Wave Simulated Annealing solver, D-Wave Hybrid solver y D-Wave Advantage6.1 QPU.

➢ D-Wave Simulated Annealing: En el caso de la implementación de este solver, la búsqueda se realiza muchas veces. Esto se denomina muestreo, y es habitual realizar desde cientos hasta varios miles de muestreos (en nuestro caso, muestreamos 1000 veces y, en algunos casos, 5000 veces). Una vez finalizado, normalmente se retienen las N mejores soluciones. A partir de ellas, se filtran las soluciones inviables.

Ejecutamos este solver en hardware común: un portátil Windows-10 con CPU IntelCorei7-1065G7 a 1,30GHz y 12,0GB de RAM y utilizando la configuración por defecto del solver.

➢ D-Wave Quantum Advantage 6.1: En nuestro caso, realizamos 1000 muestreos. La conectividad de nuestro problema es muy exigente: sólo los problemas con menos de 10 nodos pueden someterse a esta QPU. Por lo tanto, intentamos resolver sólo los escenarios de 5 nodos.

➢ D-Wave Hybrid: En nuestro caso, no establecemos un límite de tiempo, y el solver establece un valor de 3 segundos, para todos los escenarios, incluso para los escenarios con 30 nodos. En un caso aumentamos el límite de tiempo a 5 segundos, para permitir al solver alcanzar la solución óptima que no se encontró utilizando el límite de tiempo por defecto.

➢ Gurobi Classical Linear: La configuración del solver para resolver este problema se mantuvo como estándar. Además, los análisis se realizaron en una Máquina Virtual Linux 5.10.0-21-cloud-amd64 con 346 GB de RAM y 32 CPUs. Para mitigar cierto sesgo computacional, cada análisis se realizó 1000 veces, de modo que el tiempo transcurrido es la media de todas estas muestras.

Una vez obtenidas y tratadas las soluciones, vamos a representar gráficamente los resultados de cada una de las formulaciones. Para ello, calculamos la relación de energía entre las energías de las soluciones con respecto a las energías óptimas esperadas, para cada solver. Para las soluciones factibles, esto muestra lo lejos que están de las soluciones óptimas.

5.1.2.1 FORMULACIÓN CUADRÁTICA

Comenzamos implementando la formulación cuadrática y una vez probado el modelo, procederemos a resolver los escenarios de datos mencionados anteriormente obteniendo los siguientes resultados.

En primer lugar, representamos gráficamente la relación de energía en función del número de nodos.

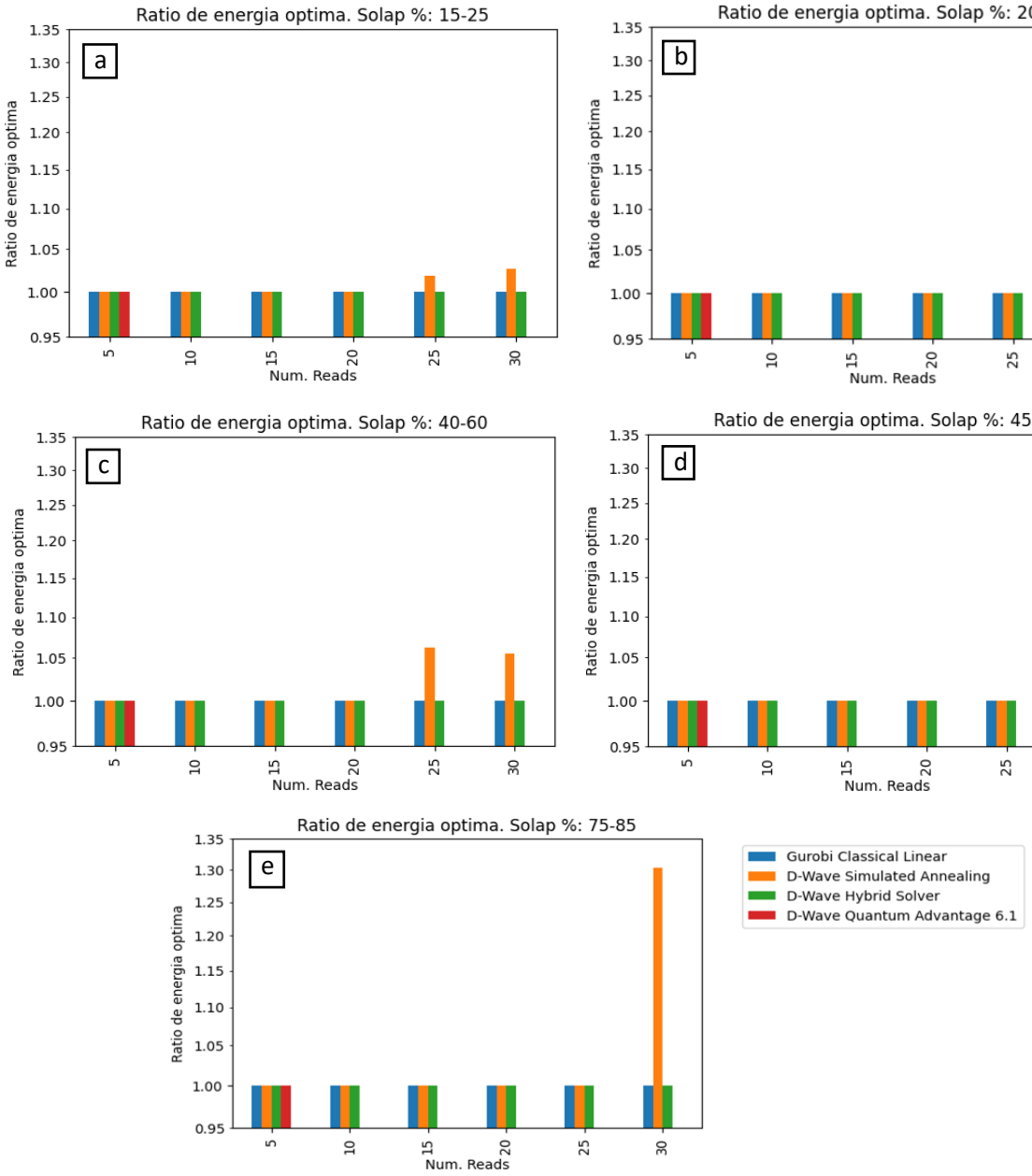


Figura 4. Relación de energía en función del número de nodos para solapamientos entre a)15-25%, b)20-80%, c)40-60%, d) 45-55%, e)75-85%

La figura 4 muestra cómo el solver D-Wave Hybrid y el solver Gurobi Linear han obtenido soluciones óptimas (el ratio es 1.0) para todos los escenarios propuestos con todos los rangos de solapamiento.

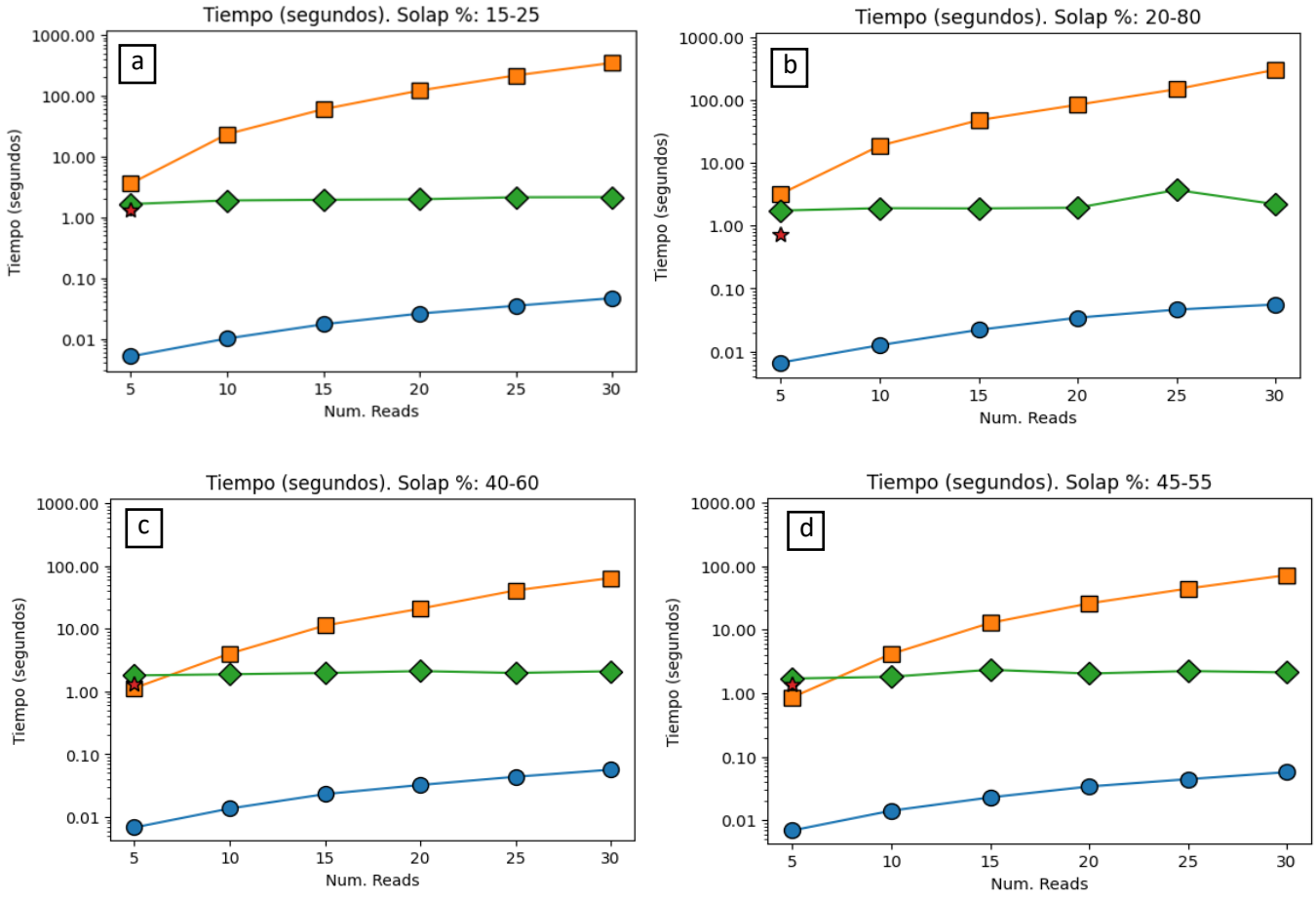
Sin embargo, el solver D-Wave Advantage 6.1 sólo ha sido capaz de obtener soluciones óptimas para los casos de cinco nodos, con 1000 muestras por ejecución y sólo soluciones inviables (por lo que no se muestran en las figuras) para los casos de más nodos. Hasta cierto punto, esto era de esperar,

debido a la limitada capacidad para codificar este tipo de problemas en las QPU de D-Wave, por el número de variables y la alta conectividad requeridas. Como ya se ha explicado, los escenarios con más de 10 nodos superan el límite de este tipo de QPU.

El solver Simulated Annealing obtiene soluciones óptimas para los escenarios de bajo número de nodos. Pero a medida que aumenta el número de nodos (25, 30), se empiezan a obtener soluciones factibles (no óptimas) en la mayoría de los casos. Cabe señalar que aún existen algunas soluciones óptimas para solapamientos elevados. La figura 4 b), d) y e) muestran cómo para 25 nodos hay soluciones óptimas, correspondientes a los solapamientos, (20-80, 45-55, 75-88,), respectivamente. Para 30 nodos, analizando todas las Figuras anteriores, ya no se encuentra una solución óptima, sino una factible. Ese resultado indica que es más fácil para el solver Simulated Annealing realizar el ensamblaje del genoma cuando los solapamientos entre reads son mayores. Sin embargo, no se puede concluir definitivamente, ya que habría que explorar soluciones para un mayor número de nodos.

De nuevo, la escala y la complejidad del problema entran en juego, pero a diferencia del solver D-Wave Advantage 6.1, el algoritmo Simulated Annealing puede encontrar soluciones óptimas en escenarios con hasta aproximadamente 25 nodos.

A continuación, se han comparado los tiempos de ejecución de cada solver en función del número de nodos o lecturas.



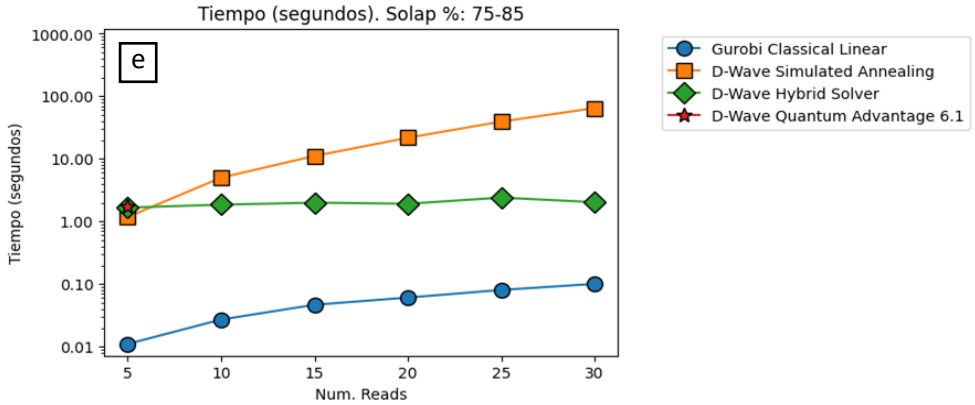


Figura 5. Tiempo de ejecución en función del número de nodos para solapamientos entre a)15-25%, b)20-80%, c)40-60%, d) 45-55%, e)75-85%

Analizando los tiempos de computación, se puede observar en la figura 5, la formulación lineal generada con Gurobi es más rápida que el resto de solvers. Hay que tener en cuenta, como hemos mencionado, que la aproximación clásica utilizada aquí no es una formulación cuadrática, lo que reduce la capacidad computacional y simplifica el problema. Pero es un buen indicador de lo que se puede esperar como máximo del mejor de los mundos.

También es interesante observar que el solver D-Wave Hybrid mantiene una tendencia constante a medida que aumenta el número de nodos, mientras que con Gurobi, la tendencia aumenta linealmente con el número de nodos. Se trata de un resultado digno de mención porque plantea la cuestión de qué ocurrirá cuando el número de nodos se incremente por encima de 30. Queda por estudiar si esta tendencia se mantiene o si el solver híbrido supera el tiempo de cálculo del enfoque clásico.

Por otro lado, el solver Simulated Annealing muestra una tendencia de tiempos crecientes con el número de nodos muy similar a la del solver Lineal de Gurobi, salvo, claro está, dos órdenes de magnitud por encima de éste. El aumento de los tiempos de duración entre los casos de 5 nodos y 30 nodos es de dos órdenes de magnitud en el solver de Simulated Annealing (y de sólo 1 orden de magnitud en el solver Lineal de Gurobi). Además, varía en función del grado de solapamiento. Por ejemplo, en las figuras 5 a) y b) para 5 nodos, el tiempo es de unos segundos. Sin embargo, a medida que aumenta el número de nodos y el tiempo, se alcanzan los 30 nodos con un tiempo de cálculo que supera los 100 segundos. Comparando este rango con las figuras c), d) y e), se observa que, para 5 nodos, el tiempo también es de unos pocos segundos, pero cuando se alcanzan los 30 nodos, el tiempo no supera los 100 segundos. Si se observan los intervalos de solapamientos en los que esto ocurre, el tiempo de cálculo para encontrar una solución óptima aumenta a medida que aumenta el número de nodos para solapamientos de intervalos amplios (20-80) y bajos (15-25). De hecho, aunque pedíamos 1000 disparos o muestras en cada ejecución del algoritmo de simulated annealing, eso no era suficiente en el caso de estos escenarios más complejos con estos solapamientos. Por lo tanto, tuvimos que aumentar el número de muestras a 5000 para obtener soluciones óptimas o incluso sólo factibles. Esto justifica el aumento del tiempo en un factor de aproximadamente 5 en estos casos. Este resultado nos permite verificar que para solapamientos más bajos y de rango mayor (20-80 y 15-25%) a medida que aumenta el tamaño del problema es más costoso encontrar una solución. Concluimos con que el grado de solapamiento que más beneficia la obtención de una solución son intervalos intermedios (40-60 y 45-55%) e intervalos grandes (75-85%).

Este parámetro, como hemos mencionado, tiene algunas peculiaridades que afectan a la comparación de las soluciones dependiendo de cada solver.

- D-Wave Simulated Annealing. Los tiempos de cálculo con el Simulated Annealing dependen del número de disparos/muestras solicitados. Se puede calcular un tiempo por muestra, y da una idea del coste de obtener una única solución. Pero para garantizar que se obtienen soluciones óptimas o suficientemente buenas, debe ejecutarse un número mínimo de disparos, lo que afecta al tiempo de cálculo necesario.

  Más concretamente, dado que el Simulated Annealing se ejecuta en CPU, se ve afectado por el crecimiento exponencial de la complejidad del problema cuando aumenta el número de nodos. Esto significa que, para problemas mayores, se necesita un ordenador realmente potente.

- D-Wave Quantum Advantage 6.1, Los tiempos dependen también del número de disparos/muestras solicitados. El proceso físico de quantum annealing tarda unos 20 microsegundos. El tiempo extra para el preprocesamiento del trabajo del problema y el postprocesamiento de las soluciones a devolver (procesos que se ejecutan en dispositivos D-Wave delante de la QPU) se suman a eso, hasta la escala de los milisegundos. Y esto debe multiplicarse por el número de disparos, que es del orden de 1000.

  En general, la escala de tiempo para un trabajo con 1000 disparos o muestras es del orden de algunos segundos. Pero no experimentará el crecimiento exponencial. Con este enfoque, la resolución de un problema implica una duración de tiempo bastante constante por disparo (la duración del ciclo de annealing) ampliada por los tiempos de preprocesamiento y postprocesamiento.

- D-Wave Hybrid. En cuanto al solver híbrido, el límite de tiempo que rige este enfoque es el principal factor restrictivo. En nuestros escenarios observamos que el valor por defecto de 3 segundos es suficiente para llegar a la solución óptima, y se mantiene constante para todos los escenarios. Este valor es el establecido por defecto por el propio solver según su propio criterio en función del tamaño del problema. Sin embargo, los tiempos medidos son de unos 2 segundos de media, y no superan este valor límite (salvo en un caso, con 3,7 segundos).

  Además, intentamos establecer valores de límite de tiempo más bajos y ver si se seguían obteniendo buenas soluciones. Sin embargo, el solver no permite establecer límites de tiempo inferiores a los sugeridos por él mismo, en función del tamaño del problema, por lo que tuvimos que ceñirnos a 3 segundos en todos los casos.

  Cabe esperar que los problemas de mayor tamaño requieran aumentar ese límite de tiempo. Dado que este solver parece encontrar las soluciones óptimas con cierta facilidad, al menos en todos los casos hasta 30 lecturas, más adelante exploraremos su rendimiento para casos más grandes, hasta 100 nodos o lecturas, y ver cómo esto afecta al límite de tiempo necesario para encontrar una buena solución. Los resultados preliminares son muy prometedores, y será muy interesante ver el comportamiento del solver de aproximación clásica para estos mismos casos.

- Gurobi Classical Linear. A partir de los resultados presentados en la figura 4 es posible comprobar que el enfoque con Gurobi siempre obtuvo la solución óptima. Asimismo, en la

figura 5, se evalúa la influencia del número de nodos en el tiempo transcurrido para alcanzar la solución óptima del problema. En primer lugar, es importante observar que, en el peor de los casos, se necesitó un tiempo transcurrido bajo, en torno a 0,1 segundos, para obtener la solución óptima. Además, se observa que el tiempo necesario crece linealmente a medida que aumenta el número de nodos. Este comportamiento puede ser una tendencia natural del sistema debido a la formulación lineal utilizada, o puede ser que aún no hayamos evaluado un número significativo de nodos para poder verificar otro comportamiento.

Cabe destacar que los resultados obtenidos con el enfoque clásico no son totalmente comparables a los enfoques cuánticos, como se ha mencionado anteriormente, debido a la diferencia entre las formulaciones, una es completamente lineal (clásica), y la otra es una formulación cuadrática, pero se trata de encontrar cuál de las formulaciones con un enfoque cuántico es mejor en comparación con la mejor clásica, que es la lineal.

5.1.2.2 FORMULACIÓN LINEAL

Mediante la formulación cuadrática hemos logrado abordar problemas con hasta 30 nodos, lo que proporcionó un espacio de análisis inicial y nos ha permitido evaluar diversos solvers en cada uno de los escenarios. Esta formulación tiene limitaciones en términos de escalabilidad, lo que nos lleva a desarrollar la formulación lineal.

Con la formulación lineal hemos sido capaces de resolver problemas de mayor tamaño, con hasta 100 nodos. Este aumento del tamaño del problema, nos ha permitido ampliar el alcance de análisis y conocer de manera más completa cómo los solvers responden a desafíos más complejos.

Como vimos en los resultados de la formulación cuadrática, ciertos solvers, como el Quantum Advantage 6.1 no son capaces de obtener una solución ni siquiera factible para tamaños mayores a 5 nodos y otros como el Simulated Annealing, obtienen soluciones óptimas hasta un tamaño determinado de 25/30 nodos y a partir de estos valores, son soluciones factibles.

Estos resultados obtenidos en la formulación cuadrática nos han llevado a descartar la ejecución de estos solvers en el nuevo escenario de datos con la formulación lineal, ya que conocemos de antemano que no se van a obtener soluciones comparables con otros solver como el Hybrid o Gurobi.

Esto implica que vamos a comparar únicamente las soluciones obtenidas con el solver Hybrid y Gurobi ya que son los resultados que nos proporcionan una óptima comparación.

Este proceso de evolución en la metodología del problema destaca la importancia de adaptar enfoques según los desafíos y comparaciones que estemos dispuestos a afrontar. La transición de una formulación cuadrática a una lineal no sólo permitió la resolución de problemas más grandes, sino que también ha revelado diferencias significativas en el rendimiento de los solvers.

Dada la base de datos presentada anteriormente vamos a aumentarla adaptándola a nuestras necesidades, es decir, hasta 100 nodos. El procedimiento es el mismo a diferencia que presentamos mayor número de escenarios, además de los anteriores, añadimos tres escenarios más, 50,80 y 100 nodos. A continuación, presentamos la tabla con los escenarios añadidos a los anteriores.

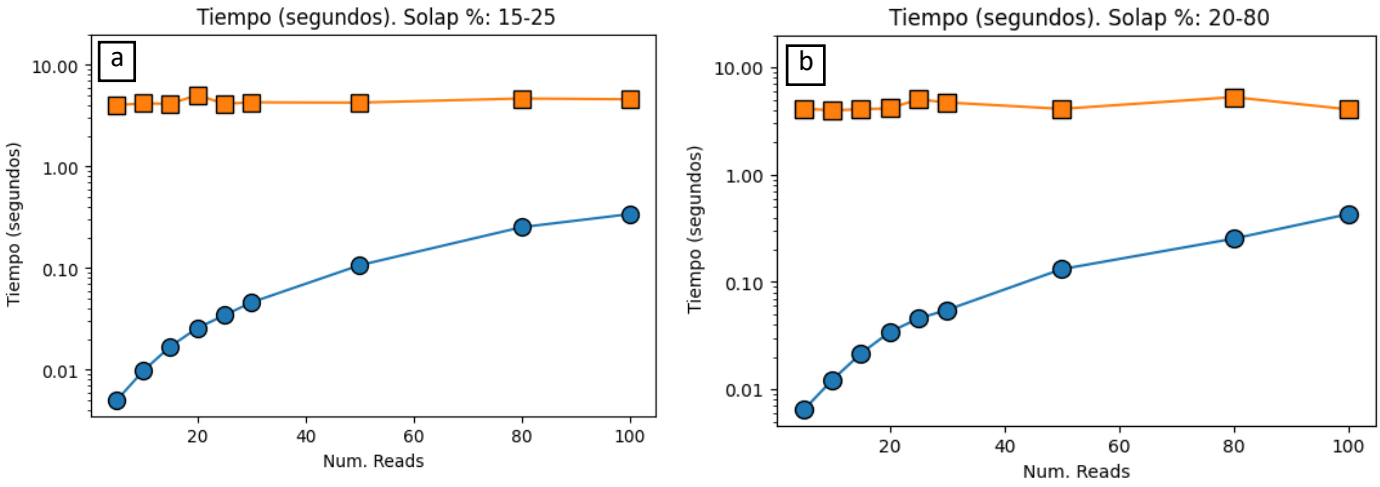
Longitud	Num. Reads	Longitud reads	Min. solap %	Max. solap %	Energía
5386	050	0134	15	25	0.81743
5386	080	0083	15	25	0.82226
5386	100	0066	15	25	0.82678
5386	050	0209	45	55	0.52577
5386	080	0131	45	55	0.52134
5386	100	0105	45	55	0.51965
5386	050	0491	75	85	0.22806
5386	080	0320	75	85	0.21580
5386	100	0254	75	85	0.21645
5386	050	0233	20	80	0.47293
5386	080	0219	20	80	0.52981
5386	100	0109	20	80	0.50099
5386	050	0212	40	60	0.51861
5386	080	0132	40	60	0.51747
5386	100	0108	40	60	0.50536

Tabla 2. Base de datos generada, añadida a la mostrada en la tabla 1

Procederemos a resolver los escenarios de datos mencionados en la Tabla 1 y los escenarios ampliados de la Tabla 2 y exponemos a continuación los resultados.

En primer lugar, en el caso anterior comparábamos el ratio de energía y lo representamos en función del número de nodos ya que había algunos solvers, de los que estábamos estudiando, que no siempre obtenían una solución óptima a diferencia del híbrido y Gurobi que sí la obtenían. En este caso, es importante destacar que con ambos solvers (Hybrid y Gurobi) hemos sido capaces de nuevo de obtener una solución óptima para todos los escenarios de datos.

En segundo lugar, el parámetro del tiempo de ejecución nos muestra la comparación determinante en este estudio. Se han comparado los tiempos de ejecución de cada solver en función del número de nodos o lecturas.



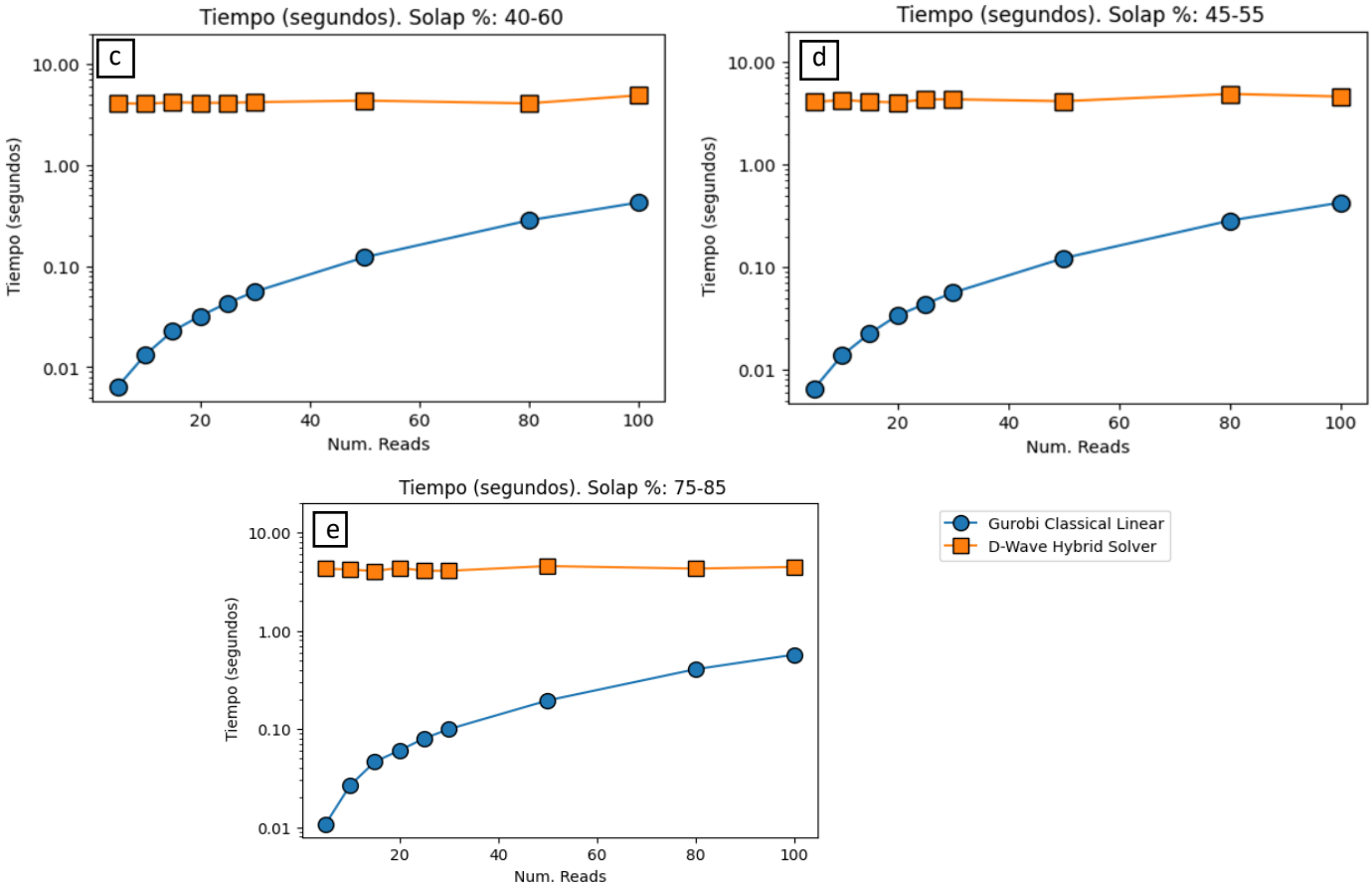


Figura 6. Tiempo de ejecución en función del número de nodos para solapamientos entre a)15-25%, b)20-80%, c)40-60%, d) 45-55%, e)75-85%

El resultado que hemos observado se presenta como un factor determinante en este estudio. En el apartado anterior, habíamos identificado que, al aumentar el número de nodos, el solver Hybrid mantenía una tendencia constante. Al ampliar nuestros datos hasta los 100 nodos, volvemos a encontrar esta consistencia en los tiempos de resolución por parte del solver Hybrid (Figura 6).

Por otro lado, previamente, habíamos notado que Gurobi seguía una tendencia lineal ascendente al aumentar el número de nodos y ahora es interesante observar (Figura 6) que la tendencia tiende de nuevo a ser lineal ya que estamos representando los ejes en escala logarítmica.

Como hemos mencionado anteriormente, el límite de tiempo en el solver híbrido es un factor restrictivo donde el valor por defecto es de 3 segundos y al aumentar el número de nodos nos damos cuenta que necesita un valor mayor a 3 segundos para llegar a la solución óptima. Los tiempos medidos son de unos 4,3 segundos de tiempo (salvo un caso, con 5,29 segundos).

A pesar de que la formulación lineal en el solver híbrido tiene un promedio de tiempo mayor en comparación con la formulación cuadrática, nuestra investigación revela un factor crucial y es que la formulación lineal permite resolver problemas de mayor magnitud con soluciones óptimas y manteniendo esa tendencia temporal constante.

Llegamos a la conclusión de que la formulación lineal se presenta como la opción más idónea para una implementación efectiva de la computación cuántica en el abordaje de este problema.

Estos resultados respaldan la idea de que el análisis de la complejidad al aumentar el tamaño del problema es fundamental para una implementación efectiva de la computación cuántica en este problema de ensamblaje de genoma.

5.2 SARS-CoV-2

Después de un análisis exhaustivo de ambas formulaciones en el contexto del ensamblaje de genomas, nuestra investigación ha trascendido hacia nuevas fases de desarrollo. Durante este proceso, surgió una estrategia que nos permitió reducir la complejidad del problema de manera significativa. En este apartado, repetiremos un análisis de nuevo con la formulación lineal, pero con un nuevo genoma, el SARS-CoV-2, responsable del Covid-19 con un tamaño de ~29.9 kbp.

Al considerar la posibilidad de aplicar un enfoque de filtrado en los solapamientos entre lecturas, abrimos la puerta a la optimización de nuestro enfoque. Aprovechando la naturaleza sintética de nuestra base de datos generada y nuestra comprensión de los rangos de solapamiento, implementamos un filtro que nos permitió descartar solapamientos que caían fuera de los parámetros definidos.
Por ejemplo, al establecer un rango de solapamiento en los datos entre el 40% y el 60%, tuvimos la capacidad de descartar conexiones que se encontraban en el 15%. Este proceso de filtrado resultó en una reducción significativa del número de variables involucradas, lo que a su vez optimizó la formulación lineal que nos permitía reducir las conexiones entre nodos. Al eliminar conexiones que no cumplían los requisitos de solapamiento, logramos una simplificación efectiva del problema y una mejora en la eficiencia de los algoritmos de resolución.

El objetivo de este apartado es encontrar cuál es la capacidad máxima que es capaz de resolver el solver Hybrid, que es el que estamos estudiando más a fondo, ya que es el único del software de D-Wave Systems que es capaz de encontrar soluciones óptimas para un mayor número de nodos.

5.2.1 BASE DE DATOS

Al igual que con los enfoques anteriores, es necesario crear una base de datos sintética con escenarios para ejecutar. En este caso, vamos a basarnos en los mismos parámetros que anteriormente, pero vamos a crear una base de datos con escenarios que tengan mayor número de nodos.

En este genoma, vamos a analizar únicamente solapamientos entre 40 y 60% ya que hemos visto que es el más idóneo y correcto para obtener soluciones y haremos un barrido desde los 5 hasta los 5000 nodos. Dentro de esta base de datos, además de los parámetros anteriores desarrollados, vamos a incluir dos parámetros más que determinan la capacidad del problema.

- Num_var: Número de variables del problema, es decir, todas las conexiones entre nodos dentro del grafo dirigido.

- Num_var_red: Número de variables reducidas, hace referencia a las variables finales del problema tras filtrar aquellas que no tienen significancia de solapamiento.

Longitud	Num. Reads	Longitud reads	Min. solap %	Max. solap %	Energía	Num_var	Num_var_red
29903	0005	8983	40	60	29903	00000020	00020
29903	0110	4928	40	60	29903	00000090	00090
29903	0015	3544	40	60	29903	00000210	00210
29903	0020	2687	40	60	29903	00000380	00380
29903	0025	2192	40	60	29903	00000600	00600
29903	0030	1844	40	60	29903	00000870	00870
29903	0050	1141	40	60	29903	00002450	02450
29903	0080	0711	40	60	29903	00006320	06320
29903	0100	0578	40	60	29903	00009900	09900
29903	0150	0388	40	60	29903	00022350	07789
29903	0200	0295	40	60	29903	00039800	01339
29903	0250	0236	40	60	29903	00062250	02016
29903	0300	0197	40	60	29903	00089700	08925
29903	0350	0168	40	60	29903	00122150	11761
29903	0400	0146	40	60	29903	00159600	16011
29903	0450	0131	40	60	29903	00202050	19836
29903	0500	0117	40	60	29903	00249500	25127
29903	0600	0098	40	60	29903	00359400	10557
29903	0700	0084	40	60	29903	00489300	13528
29903	0800	0073	40	60	29903	00639200	18266
29903	0900	0066	40	60	29903	00809100	22115
29903	1000	0059	40	60	29903	00999000	08842
29903	1500	0039	40	60	29903	02248500	18752
29903	2000	0029	40	60	29903	03998000	10384
29903	2500	0023	40	60	29903	06247500	15184
29903	3000	0020	40	60	29903	08996998	21417
29903	4000	0014	40	60	29903	15995994	12134
29903	5000	0011	40	60	29903	24994968	15783

Tabla 3. Base de datos generada SARS-CoV-2.

Analizando la tabla 3, observamos que el número de variables aumenta en gran tamaño con el número de nodos. En cambio, las variables tras reducir el problema no aumentan con el número de nodos, sino que se mantiene entre 10.000 y 20.000 variables. Esto se debe a que tras realizar varias pruebas y filtrando lo mínimo posible en cada escenario, nos hemos dado cuenta de que el solver no es capaz de resolver a partir de aproximadamente 20.000 variables. Con esto, tuvimos que reducir en torno a ese valor. El porcentaje de filtrado es desigual en cada escenario debido a que la generación de los datos conlleva un solapamiento aleatorio.

El número de variables indica el tamaño del problema, pero no es suficiente para saber si el problema se puede resolver o no. Los nodos están dispuestos formando un espacio de soluciones las cuales, no siempre son fáciles de encontrar y además del número de variables, depende de la disposición de los

nodos. Estos nodos generan una superficie de soluciones que a veces es compleja, aunque el número de variables sea bajo y aparentemente resoluble.

5.2.2 RESULTADOS

En este enfoque, con el nuevo genoma de SARS-CoV-2, vamos a analizar los mismos parámetros, energía y tiempo de ejecución.

Respecto a la energía, somos capaces de obtener la energía óptima en cada uno de los escenarios hasta 2500 nodos. El algoritmo no es capaz de resolver los escenarios de 3000, 4000 y 5000 nodos. Si observamos la tabla 3, algunos de los escenarios de 5 a 2500 nodos tienen el valor de variables reducidas mayor que el caso de 3000, 4000 y 5000 nodos, pero como hemos mencionado, obtener una solución no sólo depende del número de variables sino de la superficie de soluciones que se genere en cada escenario con el número de nodos dados.

A pesar de que el número de variables al que estoy reduciendo sea menor que otros escenarios con menos nodos, la superficie de la función de coste en estos casos en los que no se puede resolver, será más compleja. Es decir, la superficie de la función de coste estará llena de mínimos locales de donde el solver no es capaz de salir y hasta cierto número de nodos es capaz de encontrar el mínimo global (solución óptima) pero a partir de 2500 nodos la superficie es tan compleja que el solver no es capaz de llegar a esta solución óptima.

Por otro lado, representamos el tiempo de ejecución en función del número de variables, que es el parámetro que nos indica la escalabilidad del problema.

Como ya conocemos, el tiempo límite en este solver Hybrid es un parámetro que por defecto son 3 segundos, pero puede ser un parámetro regulable. En todos los escenarios, hemos dejado al solver obtener una solución sin regular el tiempo y si la solución que proporcionaba era la óptima, nos quedamos con ese valor temporal. En caso contrario, si el solver no obtenía una solución óptima, aumentamos el límite de tiempo para permitir al solver alcanzar la solución óptima que no encontraba utilizando el límite de tiempo por defecto. Esta regulación temporal nos permitía en muchos casos obtener una solución óptima al aumentar el tiempo.

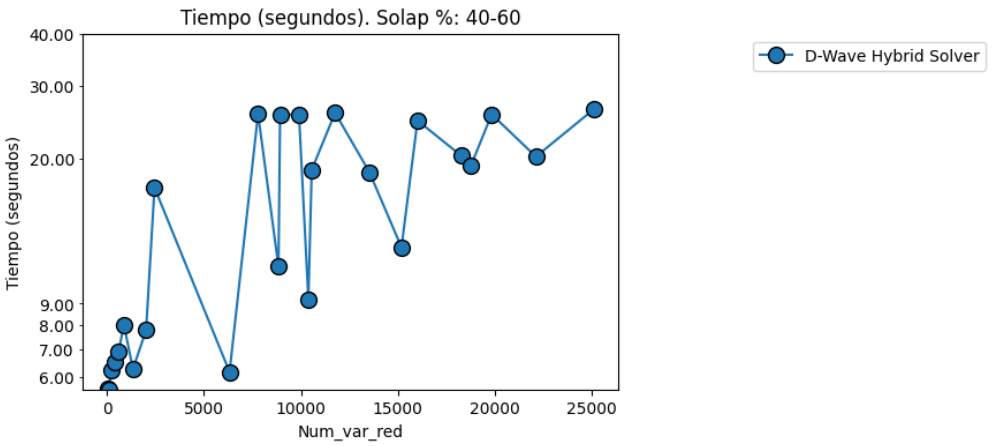


Figura 7. Tiempo de ejecución en función del número de variables reducidas para solapamiento entre 40-60%.

Cabe mencionar de nuevo que en esta Figura 7, el aumento de variables no va directamente relacionado con el número de nodos (Tabla 3). Como observamos en la Figura 7, el tiempo se ve oscilante, pero en general, se observa una tendencia ascendente con el número de variables. En los apartados anteriores la tendencia temporal era constante al aumentar el número de nodos (hasta 100), en este caso vemos una tendencia ascendente. Es importante destacar que a pesar de aumentar el número de variables el tiempo no se dispara (entre 3 y 30 segundos), desde aproximadamente 700 variables hasta 25000 hay valores temporales más bajos correspondientes a los tiempos por defecto que establece el algoritmo y si observamos esos valores en torno a 30 segundos corresponden a los escenarios en los que hemos regulado el tiempo, es decir, hemos establecido 30 segundos de tiempo fijo para que sea capaz de encontrar la solución óptima.

6. CONCLUSIONES Y PRÓXIMOS PASOS

Hemos descrito y evaluado los resultados obtenidos con los solvers en los enfoques cuántico y clásico. Esto incluye una comparación entre el rendimiento de los solvers cuánticos y clásicos. Ahora presentamos nuestras principales conclusiones sobre las ventajas y limitaciones del enfoque cuántico y se discuten posibles direcciones futuras para la investigación en este campo.

Las conclusiones con respecto a cada uno de los solvers son:

- El solver D-Wave Hybrid y el lineal clásico Gurobi han mostrado los mejores rendimientos en la obtención de soluciones óptimas para todos los escenarios propuestos.

- El solver puramente cuántico D-Wave Advantage 6.1 QPU sólo fue capaz de obtener soluciones óptimas para el caso de cinco nodos debido a limitaciones de escala.

- El solver D-Wave Simulated Annealing obtiene soluciones óptimas para escenarios con un número reducido de nodos, pero tiene dificultades para mantener la optimalidad a medida que aumenta el número de nodos.

La optimización del diseño del modelo implementando la formulación lineal, nos ha permitido reducir considerablemente el tamaño de nuestro problema y así resolver ensamblajes de genoma más complejos computacionalmente, a diferencia de la formulación cuadrática que no nos permitía resolverlos. Evidentemente, aumentando la capacidad del hardware disponible, podemos conseguir un ensamblaje de genoma más preciso y rápido, pero hasta el momento tenemos una limitación que hemos solventado con nuevos enfoques.

El análisis del genoma bacteriófago phiX174 ha desempeñado un papel crucial proporcionándonos información sobre el comportamiento de los solvers. Al evaluar cómo interactúan los escenarios de datos con diferentes solvers, hemos podido identificar características específicas que influyen en el estudio. Mediante la comparación realizada entre solvers hemos descartado aquellos que no son adecuados ni comparables con el enfoque clásico y que, por tanto, no son óptimos para resolver el problema al que nos enfrentamos de ensamblaje de genoma.

Este análisis nos ha proporcionado una perspectiva sobre cómo se comportan los solvers en diferentes contextos, con dos formulaciones distintas y cuál es su capacidad computacional. Hemos podido evaluar su eficacia en la resolución del ensamblaje de genoma y encontrar que desafíos pueden abordar de manera efectiva.

Hemos identificado que el enfoque híbrido es prometedor en la resolución de problemas de mayor envergadura. La combinación de diferentes técnicas de este solver ha demostrado ser capaz de resolver problemas más complejos y de mayor tamaño.

Un avance notable en nuestra comprensión del potencial del solver D-Wave Hybrid se produce a través del análisis del genoma SARS-CoV-2. Usamos el solver híbrido en este enfoque y con éxito determinamos el límite computacional de este solver, es decir, identificamos el punto hasta el cuál es capaz de resolver problemas de manera eficaz, teniendo en cuenta tanto su tamaño como su nivel de complejidad.

En general, los resultados obtenidos con los distintos solvers parecen prometedores en muchos aspectos, pero al mismo tiempo ponen de manifiesto la importancia de elegir un solver adecuado para el problema y la necesidad de seguir investigando para mejorar la escalabilidad de los solvers puramente cuánticos. El éxito de los resultados de este proyecto nos ha permitido realizar un artículo [16] con algunos de los resultados para dar visibilidad a esta investigación y poder continuarla en un futuro. Algunos de los datos, como ya se referencia en el artículo [16], han sido cedidos por colaboración del proyecto con NTT DATA Brazil y el Innovation Center de NTT DATA.

El alcance del artículo [16] mostró una serie de limitaciones con unos próximos pasos y con esta memoria hemos sido capaces de avanzar en la investigación permitiendo pensar en nuevas ideas y enfoques como optimizar la formulación matemática del modelo y su implementación para los solvers o poder abordar problemas con mayor capacidad computacional. A pesar de los avances, este estudio da pie a preparar próximos pasos, cómo aplicar nuestro modelo a datos reales de secuenciación genómica y otros. Por ello, proponemos posibles avances y mejoras a nuestro modelo.

- Partición de grafos mediante técnicas como Kamedias o METIS. Para resolver grafos más grandes, podemos utilizar técnicas de particionado de grafos que dividen el grafo en subgrafos más pequeños y manejables. Con estas técnicas podemos reducir la carga computacional y obtener mejores resultados.

- Mejora de la generación de secuencias simuladas. Generación de datos de forma similar a la secuenciación real, podemos comprender mejor el potencial de la computación cuántica para el ensamblaje de genomas. Esto puede lograrse incorporando modelos de error más realistas, manejando lecturas con errores y probando los resultados del ensamblaje con un conjunto de datos mayor. Para ello, podemos desarrollar un algoritmo capaz de detectar o corregir errores en las lecturas, lo que podría conducir a un ensamblaje más preciso y realista del genoma.

7. BIBLIOGRAFÍA

[1] Morey, Marcos, Ana Fernández-Marmiesse, Daisy Castiñeiras, José M. Fraga, María L. Couce, José A. Cocho. A glimpse into past, present, and future DNA sequencing. *Molecular Genetics and Metabolism*. 2013, **110**(1-2), 3–24. ISSN 1096-7192. Disponible en: doi:10.1016/j.ymgme.2013.04.024

[2] Valencia, C. Alexander, Pervaiz, M.A., Husami, A., Qian, Y., Zhang, K.. *Next generation sequencing technologies in medical genetics*. New York, NY: Springer New York, 2013. ISBN 9781461490319. Disponible en: doi:10.1007/978-1-4614-9032-6

[3] GIBBS, Richard A. The human genome project changed everything. *Nature Reviews Genetics*. 2020, **21**(10), 575–576. ISSN 1471-0064. Disponible en: doi:10.1038/s41576-020-0275-3

[4] Feynman, Richard P. Simulating physics with computers. *International Journal of Theoretical Physics* . 1982, **21**(6-7), 467–488. ISSN 1572-9575. Disponible en: doi:10.1007/bf02650179

[5] Boev, A. S. Rakitko, A.S., Usmanov, S.R. et al. Genome assembly using quantum and quantum-inspired annealing. *Scientific Reports*. 2021, **11**(1). ISSN 2045-2322. Disponible en: doi:10.1038/s41598-021-88321-5

[6] Sarkar, Aritra, Al-Ars Z, Bertels K. QuASeR: Quantum Accelerated de novo DNA sequence reconstruction. *PLOS ONE*. 2021, **16**(4), e0249850. ISSN 1932-6203. Disponible en: doi:10.1371/journal.pone.0249850

[7] Nałęcz-Charkiewicz, Katarzyna y Robert M. NOWAK. Algorithm for DNA sequence assembly by quantum annealing. *BMC Bioinformatics*. 2022, **23**(1). ISSN 1471-2105. Disponible en: doi:10.1186/s12859-022-04661-7

[8] Lucas, Andrew. Ising formulations of many NP problems. *Frontiers in Physics*. 2014, **2**. ISSN 2296-424X. Disponible en: doi:10.3389/fphy.2014.00005

[9] Cao, Y., J. Romero y A. Aspuru-guzik. Potential of quantum computing for drug discovery. *IBM Journal of Research and Development*. 2018, **62**(6), 6:1–6:20. ISSN 0018-8646. Disponible en: doi:10.1147/jrd.2018.2888987

[10] Solenov D, Brieler J, Scherrer JF. The Potential of Quantum Computing and Machine Learning to Advance Clinical Research and Change the Practice of Medicine. *Missouri medicine*. Mo Med. 2018 Sep-Oct;115(5):463-467. PMID: 30385997; PMCID: PMC6205278.

[11] Outeiral, C., Strahm, M., Shi, J., Morris, G. M., Benjamin, S. C., & Deane, C. M. The prospects of quantum computing in computational molecular biology. *WIREs Computational Molecular Science*. 2020, **11**(1). ISSN 1759-0884. Disponible en: doi:10.1002/wcms.1481

[12] Andersson, M. P., Jones, M. N., Mikkelsen, K. V., You, F., & Mansouri, S. S. Quantum computing for chemical and biomolecular product design. *Current Opinion in Chemical Engineering*. 2022, **36**, 100754. ISSN 2211-3398. Disponible en: doi:10.1016/j.coche.2021.100754

[13] COOK, William. *In pursuit of the traveling salesman: mathematics at the limits of computation*. Princeton: Princeton University Press, 2012. ISBN 9780691152707.

[14] Escherichia phage phiX174, complete genome - Nucleotide - NCBI. *National Center for Biotechnology Information*. Disponible en: https://www.ncbi.nlm.nih.gov/nuccore/9626372

[15]. Severe acute respiratory syndrome coronavirus 2 isolate Wuhan-Hu-1, co - Nucleotide - NCBI. *National Center for Biotechnology Information*. Disponible en: https://www.ncbi.nlm.nih.gov/nuccore/NC_045512.2

[16] Optimización del proceso de ensamblaje genómico basado en tecnologías cuánticas y comparativa con aproximaciones clásicas | NTT DATA Spain, NTT DATA Brazil y Centro de Innovación Cuántica de NTT DATA. Disponible en: https://es.nttdata.com/newsfolder/optimizacion-del-proceso-de-ensamblaje-genomico-basado-en-tecnologias-cuanticas