Optimization of Genome assembly using Quantum Computing

June, 2023

TABLE OF CONTENTS.

1. INTRODUCTION

Genome assembly is the process of piecing together the long and complex molecular sequences that make up an organism's genome. This process involves aligning and merging the overlapping fragments of DNA or RNA sequences that are generated during sequencing, in order to reconstruct the full genomic sequence.

This is particularly valuable in the fields of genomics, molecular biology, and personalized medicine. By obtaining a complete genome, researchers can identify and analyse genes responsible for disease or traits, develop new treatments and therapies, and better understand the interactions between genetic and environmental factors. Additionally, genome assembly can accelerate drug discovery by providing a complete roadmap of an organism's genetic material, which can help researchers identify potential drug targets more efficiently. Overall, genome assembly is a critical tool for advancing knowledge and innovation in the Life Sciences and Health sectors.

Quantum computing has the potential to accelerate genome assembly by providing new algorithms and computational methods that can operate on large datasets with greater speed and efficiency. Traditional computing methods rely on brute force algorithms to solve complex mathematical problems, which can be computationally expensive and time-consuming. In contrast, quantum computers can perform complex calculations much faster and more efficiently, helping researchers to process larger amounts of genomic data in less time.

Quantum computing can also support genome assembly by enabling researchers to explore novel approaches to genetic sequencing and analysis. For example, quantum algorithms could be used to analyse the unique patterns and characteristics of genetic data, allowing researchers to identify specific genes or genetic variations with greater precision and accuracy.

Overall, quantum computing has experienced a significant boom in the last decade, which has enabled to address problems that were previously impossible to solve in the classical paradigm. Thus, quantum computing holds great promise for advancing our understanding of genomics and revolutionizing the way we approach genetic research and analysis. As the technology continues to develop, we can expect to see even more innovative applications and solutions emerge in the field of bioinformatics.

In this project, – Optimization of Genomic Sequencing using Quantum Computing – we aim to explore the capability and feasibility of using Quantum Computing for genome assembly by benchmarking quantum and non-quantum computation approaches while addressing a genomic assembly problem using the same simulated real-world inspired.

The objectives we aimed to achieve included:

1. Implement classical (non-quantum) algorithms as a baseline for reference. This approach has been based on classical solvers (like Gurobi).

2. Implement a quantum computing algorithm for genome assembly, using quantum computing hardware and software to develop and implement a quantum algorithm

for genome assembly. Quantum approaches include D-Wave's superconducting quantum annealer and NTT's Coherent Ising Machine (CIM), photonic-based.

3. Compare clasical and quantum approaches againts well established KPIs, namely agility, computational efficiency, accuracy and scalability.

The problem is modeled in the framewok of Graph Theory and Combinatorial Optimization Theory [1], [2], where based on the overlaps between the genome fragments, the correct order of the fragments must be found. The classical approach involves the use of heuristics as well as exact algorithms to solve this problem. The quantum approach involves heuristic algorithms to expres the problem as a Combinatorial Optimizacion problem and solve it in quantum hardware, which takes advantage of the properties of quantum mechanics to perform the same task in a potentially more efficient manner. The superposition of states and entanglement between qubits are the basis of the development of quantum computing.

This document summarizes the definition of the problem, the involved parties and roles, the activities carried out in the scope of the project, the dataset to assess the model, as well as the project outcomes, comparison of results and conclusions of each of the teams involved in the project. Last but not least, some assumptions of the model formulation, problems identified and lessons learned have also been included.

2. INVOLVED PARTIES AND ROLES

NTT DATA SPAIN is the main promoter of the project. NTT DATA SPAIN leads the global project and brings both the technical expertise in terms of quantum computing models development and the management expertise to ensure the milestones and objectives of the project are accomplished. NTT DATA includes teams belonging to the following teams and industries:

- Life Sciences Industry: Life Sciences expertise and project managmenet and coordination
- Health Industry: Health innovation expertise and project management and coordination
- SUSI - Smart Innovation & Strategic Investments: Quantum Computing experts with focus on D-Wave's superconducting quantum annealer approach.

NTT DATA BRAZIL leads the classical approach. It develops a model based on classical computing with computational optimization tools for the resolution of the genome assembly. It also participates in the rest of the tasks providing the functional and technical experience to achieve the milestones and objectives of the project.

NTT DATA Innovation Center (IC Quantum) leads the adaptation of the architecture of the quantum model provided by NTT DATA SPAIN to be run on NTT's Coherent Ising Machine (CIM). It also participates in the rest of the tasks providing the functional and technical experience to achieve the milestones and objectives of the project.

3. QUANTUM COMPUTING POTENTIAL. A LONG-TERM VISION

3.1 Quantum computing in life sciences and health industries

Quantum Computing has experienced a significant boom in the last decade, allowing to start addressing real world problems that were previously impossible or impractical to solve using classical algorithms.

Quantum optimization algorithms are particularly relevant to the healthcare sector and may offer a competitive advantage when it comes to complex computing problems as well as those requiring high computational capacity. This is the case of Genomics, where there are currently **genome optimization problems** [3],[4],[5] (such as genome assembly) whose computational complexity is rapidly increasing, and the classical computing approaches encounter great difficulties when trying to solve them. Quantum computing in this field can lead to more accurate genome assemblies and faster diagnoses of genetic diseases. In healthcare, quantum computing can also be used for medical imaging and personalized medicine. Quantum algorithms could analyze large sets of **medical images**, allowing doctors to detect anomalies and diagnose diseases more quickly. Personalized medicine involves tailoring treatments to an individual's specific genetic makeup, and quantum computing can help analyze large datasets of genetic information to identify optimal treatments. Also, **Health analytics** could be faster and more accurate in the analysis of large-scale health data sets by using Quantum computing based technologies. For instance, it may speed up the analysis of electronic health Records (EHRs) and enable to health and care professionals with more accurate analysis while identifying patterns and trends in disease outbreaks and contribute to better informed decisions on – for instance – prescription optimization and effectiveness.

In the scope of Life sciences, it has also been identified concrete impacts in which Quantum computing could make a difference. This is the case of **Drug discovery**: Quantum computing could be used to simulate the behavior of molecules and predict how they will interact with other molecules, which could accelerate the drug discovery process [6]. This could lead to the development of new treatments for diseases that are currently difficult to treat. **Precision medicine** [7] is also a discipline that could benefit from quantum computing, as it could be used to analyze large amounts of genomic and clinical data to develop personalized treatment plans for patients. This could improve the effectiveness of treatments and reduce the risk of adverse side effects. With regards **Protein folding**, Quantum computing has the potential to be used to simulate the folding of proteins, which is a critical process in understanding how proteins function and how they can be targeted by drugs [8],[9]. This could lead to the development of new treatments for diseases such as cancer and Alzheimer's.

Overall, quantum computing has the potential to accelerate scientific discovery and improve our understanding of complex biological systems, leading to new treatments for diseases and better health outcomes for patients. It represents a revolutionary paradigm shift in the field of life sciences and offers researchers and practitioners unprecedented opportunities to accelerate scientific discovery, improve our understanding of complex biological systems and unlock the mysteries of biology and bioinformatics. As technology matures and investment in quantum research continues to accelerate, the potential of quantum computing to revolutionize healthcare and drive medical innovation is truly limitless. While challenges remain, quantum computing will play a crucial role in shaping the future of healthcare and transforming the way we think about the fundamental workings of life itself.

3.2 Vision of the NTT DATA Quantum Innovation Center

The Innovation Center focuses on advanced technologies of emerging domains, which are expected to become mainstream within the next five to ten years, aiming to establish world-leading research and development team. It is the bridge to transfer technologies from the research sector to the commercial one and to support our position as first mover on the market. Our mission is threefold:

1. We believe Quantum Computing is a long-term effort.
2. To support our customers in this long-term endeavor, we develop from system integrator to innovation partner.
3. We aim to become a Quantum System Integrator, providing the capabilities to develop, integrate, operate, sustain, and scale Quantum Computing and Quantum Inspired end-2-end solutions to our customers.

To this end, our consulting strategy stands on the followings four pillars:

- **First, find your fit.**

Quantum computing will not be for everyone. But if your business is in a data-intensive industry or a sector where simulations of complex and dynamic real-world scenarios are relevant, we recommend that you start to engage with this advanced technology. A good first step is to launch an initiative to build an understanding of quantum algorithms and gain experience using the existing quantum (and quantum-inspired) platforms and tools. But if the transformative value of quantum computing is at least five to ten years away, why should we consider investing now?

- **Steep learning curve.**

First, this is a radical technology that presents daunting acceleration challenges. Both quantum programming and the quantum technology stack bear little resemblance to their classic counterparts (although the two work together closely). Early adopters achieving expertise, visibility into knowledge and technological gaps, and even intellectual property, will be put at a structural advantage as quantum computing gains commercial traction.

- **Expect sudden breakthroughs.**

More importantly, progress towards maturity in quantum computing is not expected to follow a smooth, continuous curve. Rather, quantum computing is a candidate for a sharp turnaround that can come at any time. Companies that have invested to integrate quantum computing into their workflow are far more likely to be able to capitalize quickly, and the gaps they open will be difficult for others to close. This will offer a substantial advantage in industries where classically intractable computational problems lead to bottlenecks and lack of revenue opportunities.

- **Many alternative approaches.**

Finally, although today's quantum race focuses on the realization of Universal Quantum Computers as theorized by Feynman and Deutsch in the 1980s, alternative approaches to quantum and non-Von Neumann techniques are already available on the market. They are not general purpose, but they prove effective in solving a wide range of usually intractable combinatorial optimization tasks. Also, development of "quantum-inspired" algorithms is gaining traction, in turn leading to the realization of new digital (i.e., classical) hardware architectures capable to get the best from them.

4. PROJECT DESCRIPTION

4.1 Relevance of the genome assembly problem.

A genome contains all the genetic information that makes an organism unique. Assembling the genome from the fragments or reads produced by sequencing techniques allows us to study this information in detail.

One major application of genome assembly is in the field of medical research. By sequencing and assembling the genomes of individuals, scientists can identify genetic variations that may be associated with certain diseases or conditions. This information can then be used to develop new treatments or therapies that target the underlying genetic causes of disease.

Genome assembly also plays a crucial role in fields such as evolutionary biology and ecology. By comparing the genomes of different species, researchers can learn about the evolutionary relationships between them and gain insights into how different organisms have adapted to their environments.

Genome assembly also has important applications in the fields of biotechnology and synthetic biology. This is especially relevant in the field of metagenomics, where assembling genomes for comparison can bring insight into the gene function of specific organisms that could have biotechnological applications. By accurately sequencing and assembling the genomes of organisms, scientists can better understand the genetic basis of traits and use this knowledge to develop new products or technologies. For example, genetic engineering techniques can be used to modify the genetic code of an organism, allowing researchers to create new strains with desirable characteristics, such as increased resistance to disease or improved yield of a particular product. Moreover, genome assembly can help design synthetic genomes of organisms such as bacteria that are able to produce biofuels, drugs or other bioproducts.

Overall, genome assembly is a critical tool for advancing our understanding of genetics and biology, and it has important implications for fields ranging from medicine to agriculture.

4.2 The problem that this Project aims to solve.

This Project on Quantum Genome Assembly aims to address one of the biggest challenges in genomics today: the computer-assisted assembly of fragmented genomic sequences, the so-called *reads*. Current sequencing technologies produce reads that are fragments of the genome. These reads are usually quite short (100-200 bp) which hinders their assembly. Some recent sequencers can produce longer reads (10k bp), but they tend to contain more sequencing errors.

Since it is impossible to distinguish an error from a real nucleotide, a great amount of data (coverage) is needed. This entails that assembly algorithms need to deal with a high number of reads to assembly a genome which usually takes a lot of time.

Currently with classical computing the existing limitations are:

- Computational difficulties for the assembly of large genomes.
- Low computational speed to meet a growing demand of sequencings, due to biotechnological advances.

In genome sequencing, quantum computing can improve the accuracy and speed of genome analysis, even for computationally demanding genomes.

4.3 Strategy and potential solutions

The *'de novo'* Genome Assembly is a technique for reconstructing the genome of an organism without a reference genome. As we mentioned, starting from the fragmented genome sequences called *reads*, the problem consists of finding the path that connects all the reads in the correct order and reconstructs the genome.

The genome assembly problem can be formulated as a combinatorial optimization problem in a graph, where the nodes represent the reads, and the connections between pairs of nodes depend on the mutual overlaps between the reads.

There are multiple ways to solve such optimization problem. In this specific case, we will be basing our approach on the Traveling Salesperson Problem (TSP), where the minimum total distance path that connects all the nodes, subject to some constraints, provides the correct ordering of the reads, thus allowing to assemble the original genome.

5. ACTIVITIES OF THE PROJECT

The set of tasks planned for the development of the Project are shown in the figure below:

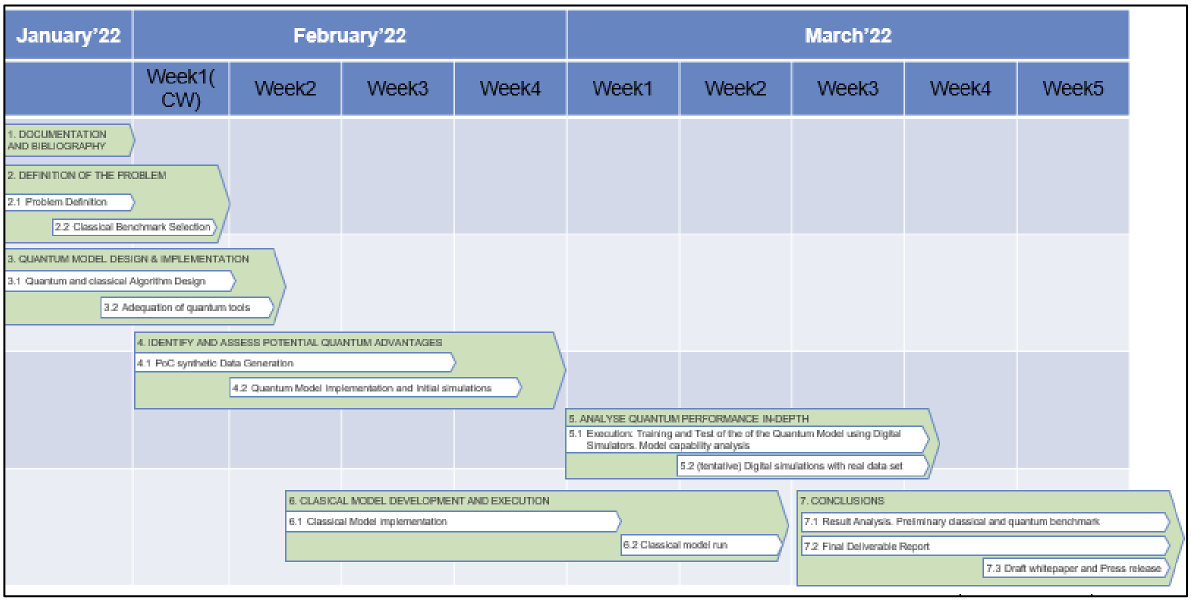


Figure 1. Project plan and activities.

5.1 Documentation and bibliography.

[3], [4], [5]

5.2 Problem definition.

This task consists of defining mathematically the combinatorial optimization problem of genome assembly, as a problem equivalent to the Traveling Salesperson Problem (TSP) and formulating it in such a way that it can be solved using quantum computing.

In genome assembly, the task is to reconstruct the original genome sequence by identifying overlapping regions in reads such as those shown below.

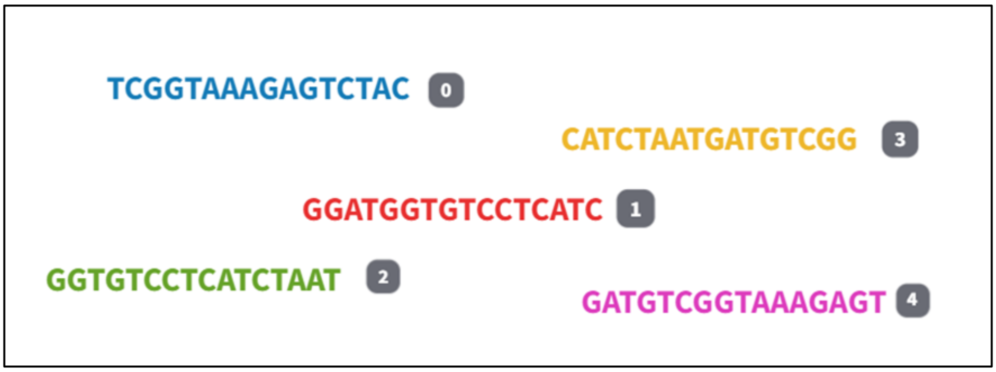


Figure 2. Example genome sequence. Reads of length 16 base-pairs.

The TSP can be modelled as a graph with nodes and edges connecting the nodes. To represent the genomic data in a graph, we create a directed weighted graph in which the nodes represent the reads, and the directed edges represent the overlap between them. This approach allows us to use graph-based algorithms to solve the assembly problem.

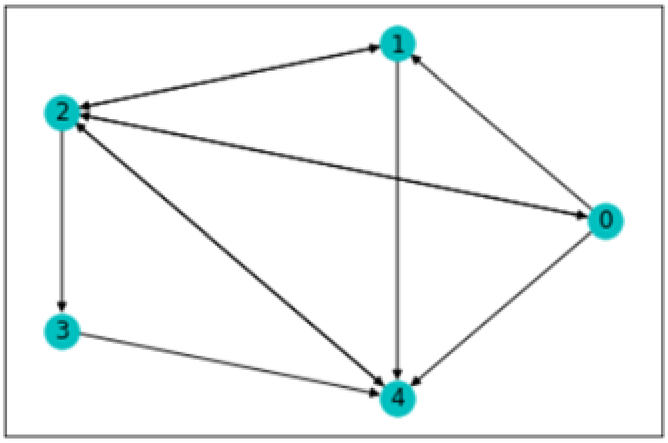
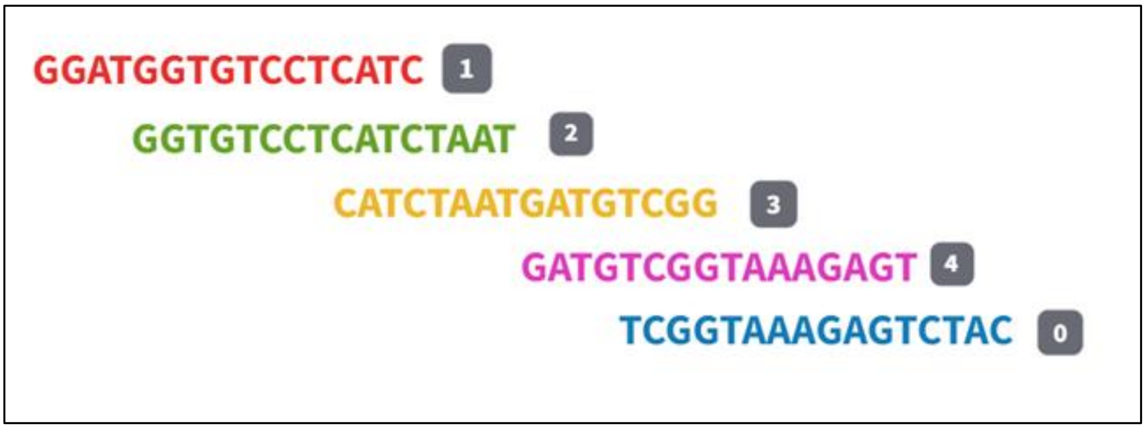


Figure 3. Directed Graph encoding the overlaps between the pairs of reads.

The overlap between two reads, i and j, which we assume are of equal length, is calculated as the number of prefix base pairs from the second read that match exactly the suffix from the first read, as can be seen in the following image.



```
GGATGGTGTCCTCATC 1
      GGTGTCCTCATCTAAT 2
                  CATCTAATGATGTCGG 3
                       GATGTCGGTAAAGAGT 4
                            TCGGTAAAGAGTCTAC 0
```

Figure 4. The example genome sequence reads, arranged to show their mutual overlaps.

We leave for future work to study variations of this and other measures, that for example allow some mismatches.

The distance between two reads can be calculated as the total length of a read minus the degree of overlap between them. That is, the greater the overlap, the smaller the distance and the closer the reads will be.

To solve the problem of genome assembly, the graph must be converted into a QUBO (Quadratic Unconstrained Binary Optimization) matrix, which is a mathematical representation of the problem that can be solved by a quantum annealer.

To get the QUBO we need to write the cost function that should be minimized (1) and the constraints that the solutions are subject to (2), (3). These are in essence the same as in the TSP.

- Cost function (1). Minimize the sum of weights (distances) along the path.

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

- Constraint (2). Each node is included in the path once and only once.

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

- Constraint (3). Each step in the path contains one and only one node.

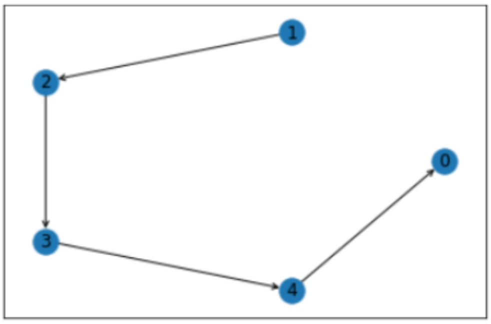
$$\forall p \in \{0...(n-1)\}, \sum_{i=0}^{n-1} x_{i,p} = 1 \qquad (3)$$

Once the QUBO is defined, the expression to minimize is given by:

$$minimize : y = x^T Q x \qquad (4)$$

Where Q is the QUBO matrix and x are the binary variables which optimize the problem.

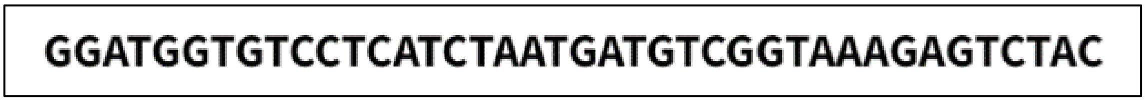
The goal is to find an assignment of binary variables (x) that minimizes the energy of the QUBO matrix, which corresponds to finding the optimal ordering of the sequence of reads. Something similar to the following image has been obtained, where the solution gives the order in which the nodes are connected to obtain the genome assembly.



```
solution
[1, 2, 3, 4, 0]
```

Figure 5. An optimal path in the Directed Graph that depicts correct order of the reads.

To finally, get the genome assembly by placing all the reads in the order indicated by the solution, i.e. by the optimal path, and taking into account that the overlaps between consecutive reads must be respected.



GGATGGTGTCCTCATCTAATGATGTCGGTAAAGAGTCTAC

Figure 6. The genome assembled, with the reads arranged in the correct order, according to their overlaps.

5.3 Quantum model design and implementation.

In this step, a quantum model for genome assembly is designed and implemented, taking into account available quantum tools, such as D-Wave.

D-Wave is a company that has been building and improving a kind of quantum computer called a quantum annealer using superconducting qubits technology. A quantum annealer is especially well suited to solve combinatorial optimization problems expressed in terms of QUBO problems. This company also provides APIs, software libraries and tools to interface with their pure quantum and hybrid quantum-classical computers.

In addition to modeling the problem, synthetic data is generated to build assembly problems and test the correctness of the quantum model developed, running small-scale problems of the genome assembly.

5.4 Coherent Ising machine design and implementation.

Quantum annealers internally map the QUBO problem into an Ising problem, where the function to minimize is the Ising Hamiltonian and the variables represent spins that can be in one of two states (up or down), represented by the values +1 and –1.

$$H = -\sum_{i,j} J_{ij}\sigma_i\sigma_j + h\sum_i \sigma_i$$

Networks of degenerate optical parametric oscillators (DOPO's) are an alternative physical system for solving the Ising problem. These are what Coherent Ising Machines (CIM) use, in this case the up-spin and the down-spin are represented by the positive and negative in-phase amplitudes (0 or π).

A CIM solves the Ising problem basing on the minimum-gain principle: if the optical couplings between the DOPO's are adjusted to implement J_{ij}, the DOPO network oscillates in the phase configuration with the lowest loss, which corresponds to the ground state of the Ising Hamiltonian. This way the CIM can provide a fully connected graph which traditional quantum annealers cannot provide.

Because of the CIM intrinsic complexity, it can be difficult to access to such technology. A possible way to overcome this limitation can be the so-called CyberCIM. CyberCIM is an early prototype of a CIM simulation performed in conventional processors.

Chaotic Amplitude Control (CAC) is an algorithm that simulate CIM through a mean-field dynamics with the addition of a control of the amplitude via error-correction terms that evolves chaotically. The algorithm was implemented using a suite for numerically solving differential equations written in Julia.

5.5 Identify and assess potential quantum advantages. Data generation.

Once the capacity and parameters that regulate the model are defined, a database is generated with different scenarios that cover several ranges of parameters that control the model.

The database we have generated comes from a sequencer simulator that we created to obtain reads from a long genome that our algorithm can solve.

Currently, in most real sequencers, many reads are generated to ensure a complete coverage of the genome, with short length reads predominating. However, for our approach, this supposed a problem since the size of the problem (number of variables) scales with N^2, where N is the number of nodes, i.e., of reads. N^2 variables scale very quickly, and this is an issue for the quantum computers that are available at the moment. To address this issue, we have generated our own simulated data. This way we can choose the number of reads and their length, making it easier to control the capacity of our algorithm and achieve accurate results.

To generate the sequences for our study, we started with the genome of the bacteriophage phiX174, which has 5386 base pairs [6]. We introduced several tunable parameters to create our dataset. Table 1 shows the data scenarios generated with the following parameters.

- First, the total genome length (Length), as mentioned above is bacteriophage phiX174.

- Second, we varied the number of reads or nodes (Num. Reads), ranging from 5 to 30 at 5-nodes intervals.
- We also adjusted the length of the reads (Reads length) to match the number of nodes needed to ensure that they overlapped enough to cover most of the genome (for convenience only; this has no effect on our approaches). It should be noted that in all cases we have covered the entire genome, length covered = 5386.
- We tested different ranges of overlaps between the reads, including low, medium and high overlap, for all node cases. The table shows the range of overlaps in percentage between the minimum (Min.Overlap %) and maximum (Max.Overlap %).
- Finally, the optimal energy refers to the minimum amount of energy (or optimal value of the cost function) that should be obtained when solving the problem, as a reference. It corresponds to the expected best solution of the graph path to assemble the genome.

Length	Num. Reads	Reads length	Min. overlap %	Max. overlap %	Energy
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

Table 1. Dataset generated.

By controlling these parameters, we can create a dataset that is tailored to our specific needs, allowing us to test the scalability and accuracy of our algorithm under different conditions.

The degree of overlap induces a varying degree of complexity into the problem, added to the scale-induced complexity (increasing number of nodes or reads). The lower the overlaps, the more similar the modeled distances between pairs of reads are, and the more difficult it becomes to find an optimal shortest path that puts the reads in the correct order.

We wanted to draw some scenarios that look realistic or achievable, and others that are less real-world cases, but represent an interesting challenge for our approaches. The meaning of each range of overlap is the following:

- 15-25 %: Very small average overlaps, very concentrated around the mean.
- 45-55 %: Moderate average overlaps, very concentrated around the mean.
- 40-60 %: Moderate average overlaps, with some spread around the mean.
- 20-80 %: Moderate average overlaps, with much more spread around the mean, with important fractions of very small and very high overlaps.
- 75-85 %: Very big average overlaps, very concentrated around the mean.

On one hand, the overlap ranges that induce more complexity *a priori* are the 15-25 and 20-80 ranges. On the other hand, the most realistic overlap ranges according to current sequencing techniques and practices are the 45-55, 40-60 and 75-85 ranges, which are also *a priori* less complex.

5.6 Analyze quantum performance in-depth.

Once the model has been tested and the final version is available, we will proceed to solving the data scenarios mentioned above. The objective is to find an optimal solution for each one, that satisfies the imposed constraints. We will show these results later, while comparing the solutions obtained by the quantum approach using D-Wave with the classical approach and the quantum approach of the CIM.

- **Quantum Annealing approach with D-Wave**. D-Wave has developed several solvers that can be applied in complex problems using quantum computing. We will run three solvers: Simulated Annealing Solver, the Hybrid Solver, and the Advantage6.1 Solver for each scenario. To understand their architecture and operation, we will explain now in some detail each of these solvers.
    - **Simulated Annealing Solver.** The Simulated Annealing Solver is an optimization algorithm that runs on *classical hardware* and seeks to find solutions to cost minimization problems. It is very similar to a hill-climbing algorithm to search for global optimal in a hypersurface defined by a cost function to be optimized. This is one of the algorithms that is used most when comparing performances of quantum and quantum-inspired approaches.
      The algorithm is in many aspects similar to the quantum annealing kind of search in the problem solutions space (but it is not a quantum annealing simulator). With the

difference that simulated annealing uses a temperature-like parameter that controls the ability to jump out of local-minima solutions found during the search, and eventually reach a global minimum (i.e., the optimal) solution. In actual quantum annealing devices, this is governed by the quantum tunneling effect that arises in physical systems at quantum scales.

The algorithm starts with a random solution candidate and, at each iteration, generates a new solution candidate that may or may not be better than the previous one, choosing a criterion based on the difference between the cost function of the new solution and the cost function of the current solution, as well as that fictitious temperature, that decreases over time.

The algorithm continues iterating until the fictitious temperature reaches a minimum value. At that moment the algorithm stops and returns the current solution as its best-found solution.

In the case of D-Wave's Simulated Annealing solver implementation, the search is performed many times. This is called sampling, and it is common to run from hundreds to a few thousand samples (in our case we sample 1000 times, and in some cases, 5000 times). After completion, the top-n best solutions are normally retained. From them, unfeasible solutions are further filtered out.

We run this solver on common hardware: a Windows-10 laptop with IntelCorei7-1065G7 CPU at 1.30GHz and 12.0GB of RAM and using the default solver settings.

- **Advantage6.1 QPU solver.** The Advantage Solver is a pure quantum solver that uses D-Wave's qubit processors (or QPUs) to solve optimization problems. These processors use an approach known as "quantum annealing" to search for the optimal solution.

  The D-Wave Advantage series of quantum computers (Advantage6.1 is their latest version) contains more than 5000 qubits and 15 couplers per qubit, or more than 35000 couplers in total. The number of qubits determines the number of problem variables that can be mapped to the processor, and the number of couplers determines how many coefficients relating pairs of variables can be represented in the QPU.

  However, due to the limited connectivity provided by this number of couplers, it is often needed to combine several qubits into groups acting as single logical qubits to match the number of quadratic coefficients of the problem. This reduces the number of qubits available for variables and limits the size of the problems that can be solved directly in a single QPU.

  Quantum Annealing operates in these processors by applying time-dependent biases and couplings to the qubits, for a duration of about 1-200 microseconds (20 microseconds is the default), resembling an annealing (or cooling) process, but quantum in nature. In the final stages, the quantum tunneling effect helps in escaping from local minima state solutions. At the end of this cycle the biases and coupling values correspond to the coefficients of the problem to be solved, and the values of the qubits encode the solution values for the variables of the problem.

  It is needed to run many samples or shots of a problem to collect sufficient statistics and ensure optimal or at least good-enough solutions are found. D-Wave enforces some limits on this number. In our case we sample 1000 times.

The connectivity of our problem at hand is very demanding: only problems with less than 10 nodes can be submitted to this QPU. Hence, we attempted to solve only the 5-node scenarios.

- **Hybrid quantum-classical solver.** The Hybrid Solver combines the power of classical and quantum computing to solve optimization problems. This approach allows us to solve many problems that do not fit on the current Advantage QPUs, due to their size (number of variables and coefficients). It can handle problems with thousands, up to a million, variables.

  This solver uses quantum-classical workflows: classical computing, to prepare an initial solution, and quantum computing, to refine and improve the solution. A time limit must be established or use a default value computed by the solver that depends on the problem size.

  The workflows combine a classical heuristic module that explores the solution space, and a quantum module which tries to solve parts of the problem in the QPU, guided by the heuristic module toward promising areas of the solution space or toward improved solutions. After the time limit, the best solution found by the heuristics is returned.

  In our case, we do not establish a time limit, and the solver establishes a value of 3 seconds, for all the scenarios, even for the scenarios with 30 nodes. In one case we increased the time limit to 5 seconds, to allow the solver to reach the optimal solution that was not found using the default time limit.

- **Classical Solver approach with Gurobi.** Gurobi is an optimizer that is state-of-the-art for solving mathematical programming problems. It is proved that the solver can provide optimal solutions for linear (LP), quadratic (QP), and mixed-integer (MIP) optimization problems. Due to its flexibility, it is possible to evaluate different approaches for the same problem with the same optimizer. To numerically solve linear programming problems, Gurobi uses the Simplex method, evaluating each iteration to see if all the constraints are satisfied and if the cost function is minimum, with a tolerance of 10^{-4}.

  The solver configuration to resolve this problem was maintained as standard. Moreover, the analyses were performed in a Virtual Machine Linux 5.10.0-21-cloud-amd64 with 346 GB of RAM and 32 CPUs. In order to mitigate some computational bias, each analysis was performed 1000 times, so that the elapsed time is the average of all these samples.

- **Quantum Inspired approach with Chaotic Amplitude Control.** NTT Basic Research Laboratories and NTT Research jointly developed the Chaotic Amplitude Control Algorithm to easily evaluate the performance of the Coherent Ising Machine in practice. Chaotic Amplitude Control (CAC) algorithm takes the optical field amplitude variables to simulate the internal process of a CIM. The time evolution of these variables is described by a sum of the interaction terms with a double well potential pot and the white noise.

  The control of the amplitude is performed by introducing error-correction terms whose role is to correct the amplitude heterogeneity. These error signals modulate the coupling strength defining a deterministic dynamic that is inclined to visit spin configurations associated with lower Ising Hamiltonian without relying entirely on the descent of a potential function.

  The time evolution equation is then numerically integrated to find the new configuration and its correspondent energy. This process is done iteratively until some condition is met, for example until it reaches a maximum number of time steps.

The value of the target amplitude for which all local minima is unstable is not known a priori, so that is dynamically modulated depending on the visited configurations to destabilize the local minima traps.

5.7 Classical model development and execution

To compare the performance of the quantum model, classical models for genome assembly were evaluated. It is possible to solve the TSP with linear or non-linear approaches and uncoupled constraints from the cost function. The non-linear approach is defined in the Problem Definition section, where the cost function is defined in (1) and is subjected to the constraints (2) and (3). Nonetheless, in this study, the classical model developed was the linear approach due to its lower complexity and typically higher performance compared to the non-linear. This approach is based on the edges of the graph, not on the nodes as the non-linear formulation, so the cost function is written as

$$\text{min:} \sum_{i=0}^{N-1} \sum_{j \neq i, j=0}^{N-1} w_{i,j}\, r_{i,j} \,,$$

where $w_{i,j}$ is the distance between the reads i and j, N is the number of reads, and $r_{i,j}$ is a binary value that is 1 if i is followed by j in the Hamiltonian path and 0 if it doesn't. It is worth pointing that r is a N-by-N matrix, where the line i corresponds to the origin and j is the destiny for given path.

The constraints for this problem must guarantee that, for each node, there is one edge going outside and one edge going inside. To do so, the first constraint is:

$$\forall j \in \{0, \dots, N-1\},\ \textstyle\sum_{j \neq i, i=0}^{N-1} r_{i,j} = 1 \,,\text{º}$$

assuring that, for each destiny node, there is only one origin node. The second constraint is:

$$\forall i \in \{0, \dots, N-1\},\ \sum_{j \neq i, j=0}^{N-1} r_{i,j} = 1 \,,$$

assuring that, for each origin node, there is only one destiny node.

Nonetheless, with only these two constraints, it is possible to achieve a solution with more than one closed path. To eliminate this, a third constraint is added as:

$$u_i - u_j + N r_{i,j} \leq N - 1 \,,\ 1 \leq i \neq j \leq N - 1 \,,$$

where u is a dummy variable array of integer values between 0 and N-1 and size N-1.

6. PROJECT OUTCOME AND LESSONS LEARNT

In this section we compare the results obtained with the different approaches used to solve the genome assembly problem. The solvers used were described above: Gurobi Linear solver, D-Wave Simulated Annealing solver, D-Wave Hybrid solver, D-Wave Advantage6.1 QPU and Chaotic Amplitude Control quantum-inspired algorithm.

Given a set of solutions obtained from the different solvers and approaches, there are several parameters and indicators than can be compared. For this PoC we focus on these two main aspects:

- Firstly, the **energy** of the solutions, i.e., the minimum energy that we have been able to find with each of the solvers and for each problem scenario.
- Secondly, the **computing time** required by each solver in each scenario. This parameter has some peculiarities that affect the comparison, and we will explain them later.

We collected solutions for all nodes and overlaps that are covered in the 30 problem scenarios with all solvers. We classified the solutions depending on their energy and whether they meet the constraints of the problem:

- **Optimal solutions**, which have an energy corresponding to the optimal energy for the specific problem scenario.
- **Feasible solutions**, which do not reach the optimal energy, but satisfy all the constraints.
- **Unfeasible solutions**, which do not meet some of the constraints, even though their energy could be lower than the optimal energy (we discard them for our analysis).

Ability to find optimal or good-enough solutions.

Reaching the optimal energy with at least one solution is important to prove that a certain approach can find the best possible solutions, given the constraints and parameters We computed the energy ratio between the solution energies with respect to the expected optimal energies, for each solver. For the feasible solutions, this shows how far they are from the optimal solutions.

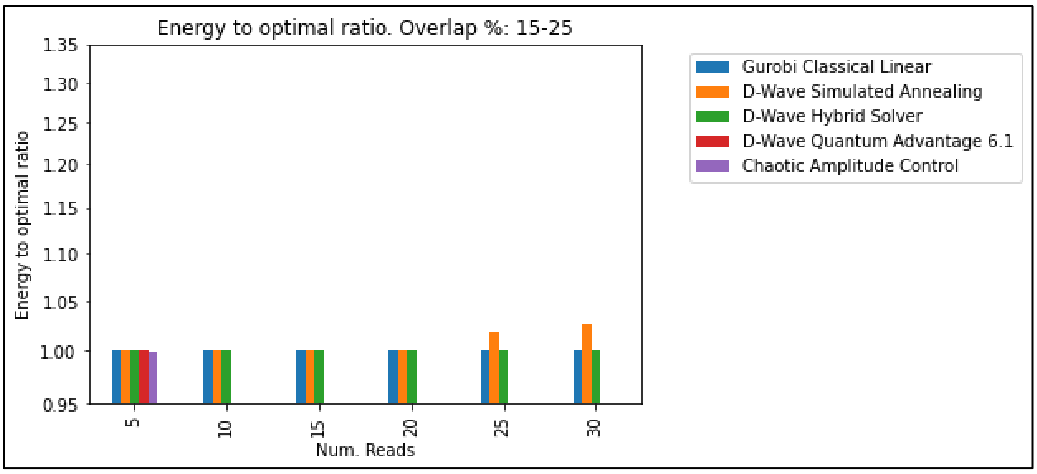


Figure 7. Energy to optimal ratio as a function of number of nodes for overlap between 15-25%.

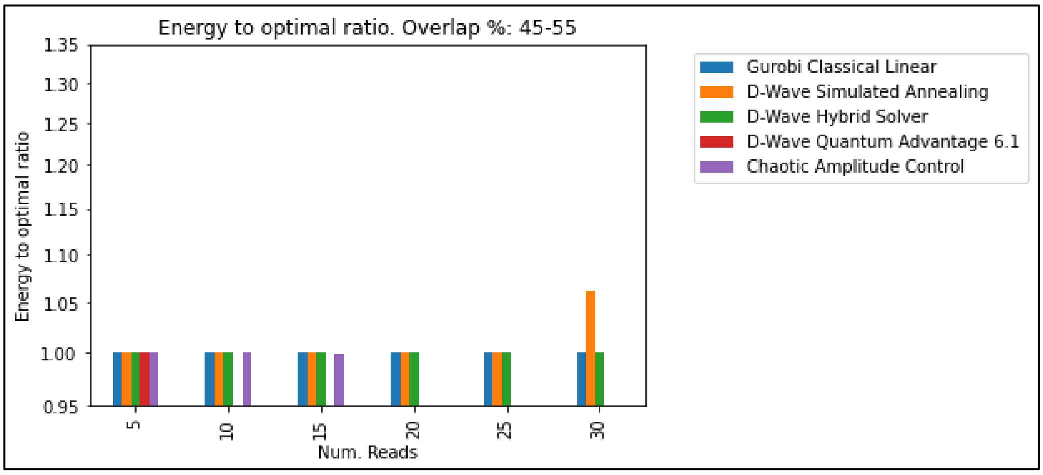


Figure 8. Energy to optimal ratio as a function of number of nodes for overlap between 45-55%.

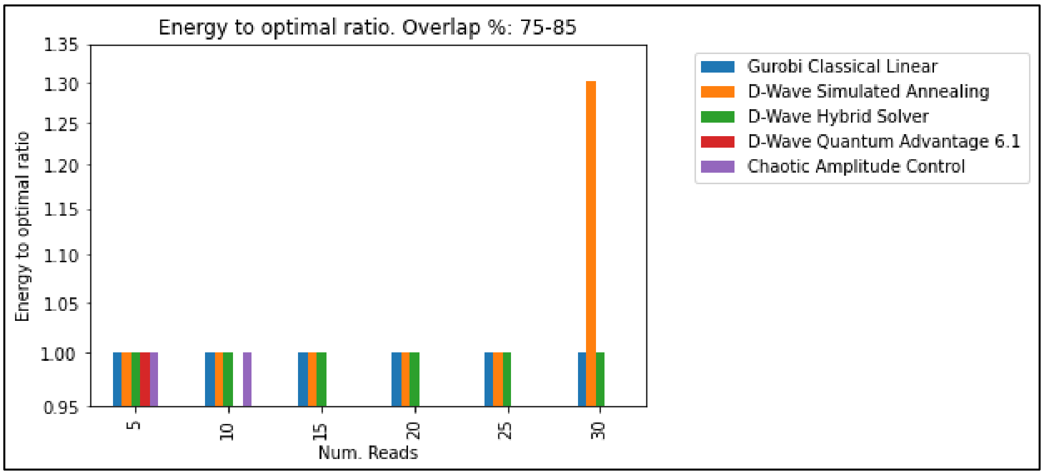


Figure 9. Energy to optimal ratio as a function of number of nodes for overlap between 75-85%.

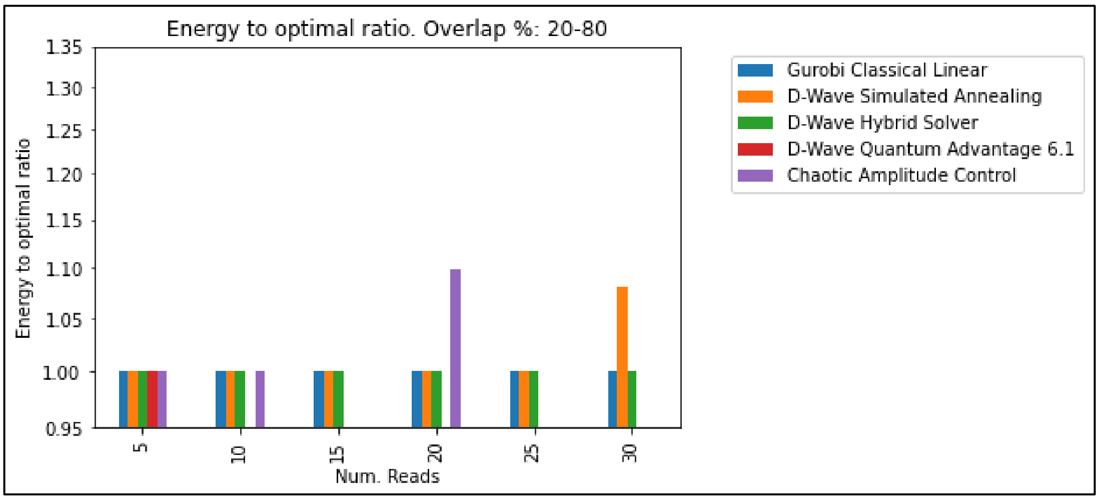


Figure 10. Energy to optimal ratio as a function of number of nodes for overlap between 20-80%.

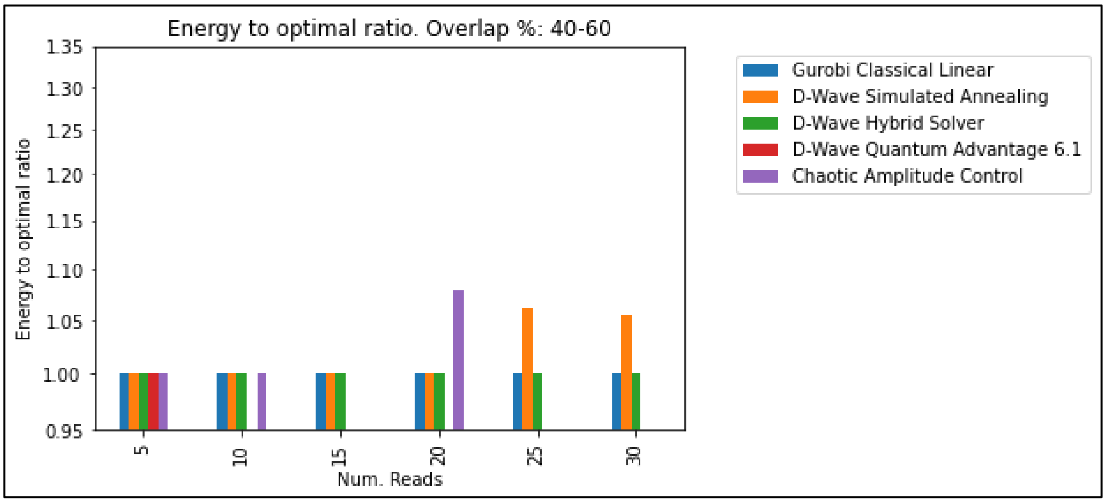


Figure 11. Energy to optimal ratio as a function of number of nodes for overlap between 40-60%.

Figures 6,7,8,9 and 10 show how the D-Wave Hybrid solver and the Gurobi Linear solver have obtained optimal solutions for all the proposed scenarios with all the overlapping ranges.

However, the D-Wave Advantage6.1 solver has only been able to obtain optimal solutions for the five-node cases, and only unfeasible solutions (therefore not shown in the figures) for the 10-node cases. As explained before, the scenarios with more than 10 nodes exceed the limit of this type of QPU, given the number of variables needed and the connectivity required. We consider these cases as if with this solver, for the scenarios from 10 nodes upwards, only unfeasible solutions were obtained.

The Simulated Annealing solver obtains optimal solutions for the scenarios of low number of nodes. But as the number of nodes increases (25, 30), we start to obtain feasible (not optimal) solutions in most cases. It should be noted that there are still some optimal solutions for high overlaps. Figures 7, 8 and 9 show how for 25 nodes there are optimal solutions, corresponding to high overlap, (45-55, 75-88, 20-80), respectively. For 30 nodes, analyzing all Figures above, no longer finds an optimal solution, but a feasible one. That result indicates that it is easier for the Simulated Annealing solver to perform genome assembly when the overlaps between reads are higher. However, it cannot be concluded definitively since solutions for a greater number of nodes would have to be explored.

Again, the scale and complexity of the problem comes into play, but unlike the D-Wave Advantage6.1 solver, the Simulated Annealing algorithm can find optimal solutions in scenarios with up to approximately 25 nodes.

Looking now at the Chaotic Amplitude Control solver in all the above figures, it is able to find optimal solutions for all overlapping scenarios but only up to 10 nodes. Other solutions obtained for 20 nodes (see Figure 9 and 10), are non-optimal feasible solutions. It is worth noting that in Figure 7 for overlaps 45-55, it has been able to obtain an optimal solution for 15 nodes as well.

Computing time and cost of finding solutions.

Next, the execution times (Wall clock time) of each solver have been compared as a function of the number of nodes or reads.

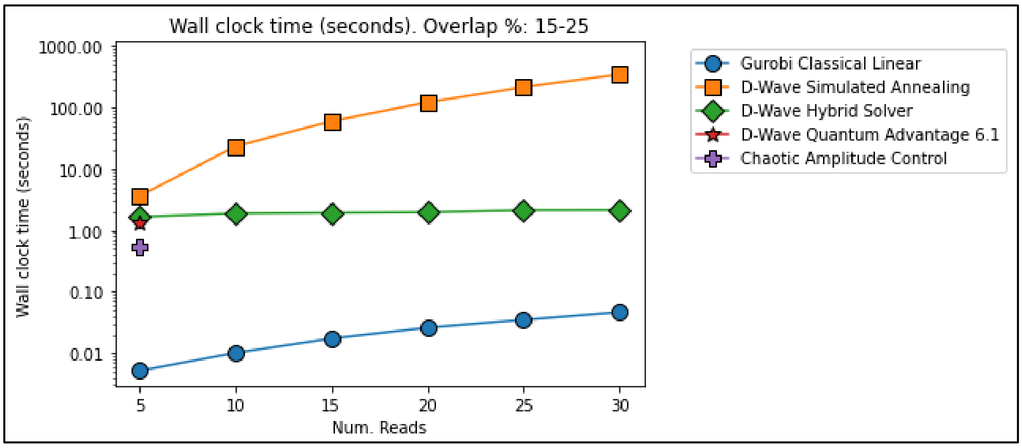


Figure 12. Wall clock time as a function of number of nodes for overlap between 15-25%.

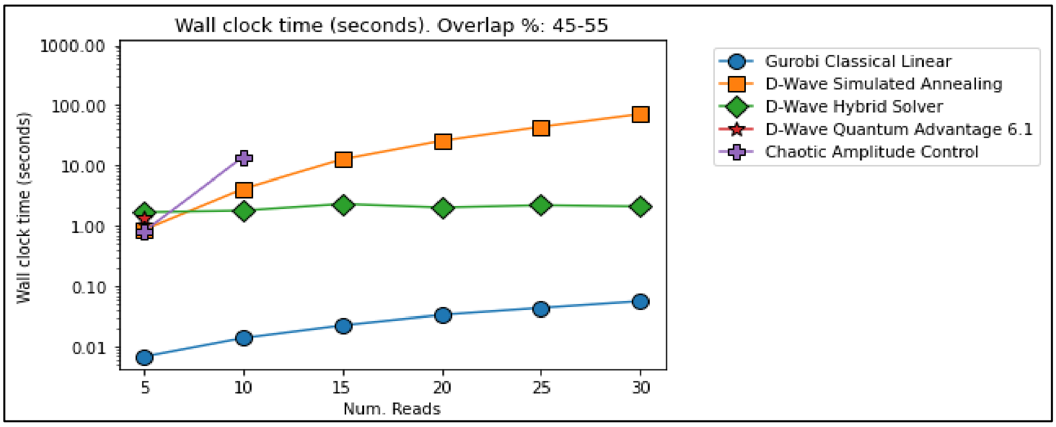


Figure 13. Wall clock time as a function of number of nodes for overlap between 45-55%.

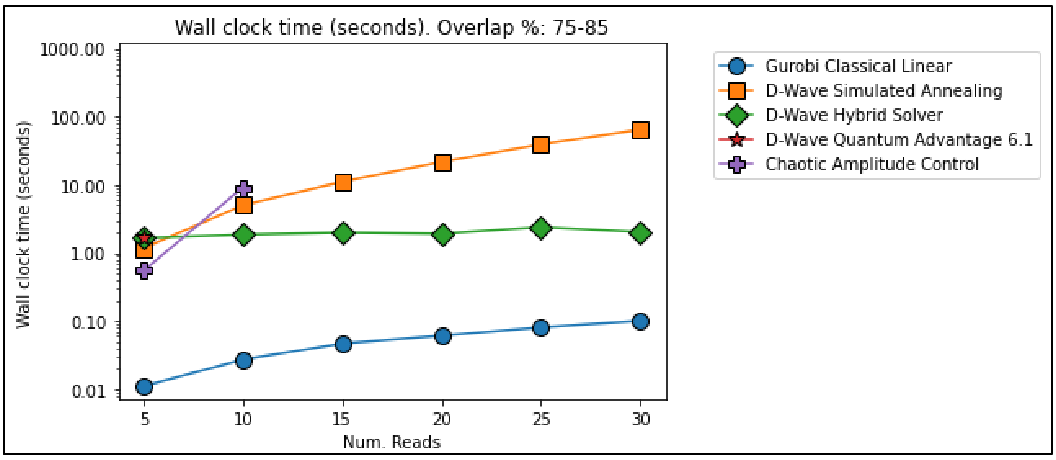


Figure 14. Wall clock time as a function of number of nodes for overlap between 75-85%.

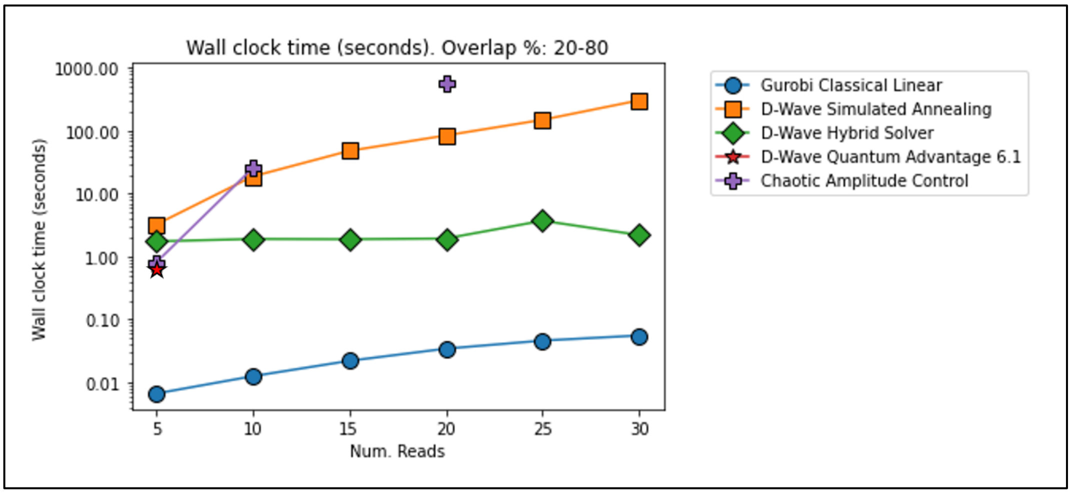


Figure 15. Wall clock time as a function of number of nodes for overlap between 20-80%.

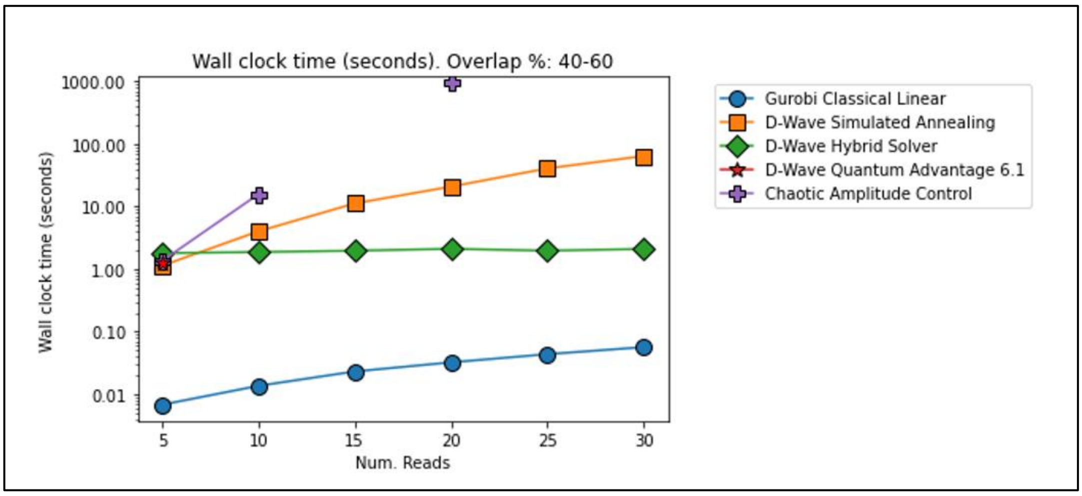


Figure 16. Wall clock time as a function of number of nodes for overlap between 40-60%.

Analyzing the computing times, it can be observed from figures 11, 12, 13, 14, and 15 that the linear formulation generated with Gurobi is faster than the rest of the solvers. It should be noted that the classical approximation used here is not a quadratic formulation, which reduces the computational capacity and simplifies the problem. But it is a good indicator of what can be expected at most from the best of all worlds.

It is also interesting to note that the D-Wave Hybrid solver maintains a constant trend as the number of nodes increases, whereas with Gurobi, the trend increases linearly with the number of nodes. This is a noteworthy result because it raises the question of what will happen when the number of nodes is increased beyond 30. It is yet to be studied whether this trend continues or if the Hybrid solver exceeds the computing time of the classical approach.

On the other hand, the Simulated Annealing solver shows a trend of increasing times with the number of nodes very similar to the Gurobi Linear solver, except of course two orders of magnitude above it. The increase in duration times between the 5 node and the 30 node cases is by two orders of magnitude in the Simulated Annealing solver (and only 1 order of magnitude in the Gurobi Linear solver). Also, it varies depending on the degree of overlap. For example, in figures 11 and 14 for 5 nodes, the time is a few seconds. However, as the number of nodes increases and the time increases, 30 nodes are reached with a computing time that exceeds 100 seconds. Comparing this range with figures 12, 13, and 15, it can be observed that for 5 nodes, the time is also a few seconds, but when 30 nodes are reached, the time does not exceed 100 seconds. If the ranges of overlaps in which this occurs are observed, the computing time to find an optimal solution increases as the number of nodes increases for wide interval overlaps (20-80) and low overlaps (15-25). In fact, while we requested 1000 shots or samples at each execution of the Simulated Annealing algorithm, that was not enough in the case of these more complex scenarios with these overlaps. Therefore, we had to increase the number of samples to 5000 in order to obtain optimal or even just feasible solutions. This justifies the increase in wall clock time by a factor of about 5 in these cases.

6.1 Quantum Annealing approach with D-Wave

The computing times with **Simulated Annealing** and with the **Advantage6.1 QPU** depend on number of shots/samples requested. A wall clock time per sample can be computed, and it gives an idea of the cost of obtaining one single solution. But to guarantee that optimal or good enough solutions are obtained, a minimum number of shots should be run, which affects the computing time required.

More specifically, since Simulated Annealing runs on CPU, it is affected by the exponential growth of the complexity of the problem when the number of nodes increases. This means that for bigger problems, a really powerful computer is needed.

In the case of the Advantage6.1 QPU, the physical process of quantum annealing takes about 20 microseconds. Extra time for preprocessing the problem job and postprocessing the solutions to be returned (processes that run on D-Wave devices in front of the QPU) add on top of that, up to the milliseconds scale. And this should be multiplied by the number of shots, which is of the order of 1000. Overall, the time scale for a job with 1000 shots or samples is of the order of some seconds. But it will not experience the exponential growth. With this approach, solving a problem implies a rather constant time duration per shot (the duration of the annealing cycle) enlarged by the pre and post processing times.

We obtained solutions with the Advantage6.1 QPU solver for the cases of 5 nodes only, with 1000 samples per run. For 10 nodes we only obtained unfeasible solutions. To some degree, this was expected, due to the limited capacity to encode this kind of problem in the D-Wave's QPUs, for the number of variables high connectivity required. An alternative, although not so common, mathematical formulation of the problem, like the one used in the Gurobi Linear approach, might help in reducing the number of variables. We leave this research for future work.

With regard to the **Hybrid solver**, the time limit that rules this approach is the main restricting factor. In our scenarios we observed that the default value of 3 seconds is enough to get to the optimal solution, and it remains constant for all the scenarios. This value is the default established by the solver itself according to its own criteria depending on the problem size. Nevertheless, the measured times are about 2 seconds in average, and do not exceed this limit value (except for one case, with 3.7 seconds). To explain this effect, either the limit time is consumed by several processes running in parallel, or the solver has a mechanism to judge whether it should stop searching and stops prematurely. We want to look deeper into this aspect in future works.

Moreover, we tried to establish lower time limit values and see if still good solutions would be returned. However, the solver does not allow to set such time limits that are smaller than its own suggested estimates, according to each problem size, and this why we had to stick to precisely 3 seconds in all our cases.

It can be expected that bigger problems will require to increase that time limit. Given that this solver seems to find the optimal solutions with some ease, at least in all cases up to 30 reads, we have started to explore its performance for larger cases, up to 100 nodes or reads, and see and how this affects the time limit required to find a good solution. The preliminary results are very promising, and it will be very interesting to see the behavior of the classical approach solver for these same cases.

6.2 Classical approach with Gurobi

From the results presented in figures 6, 7, 8, 9, and 10, it is possible to verify that the linear approach with Gurobi always obtained the optimal solution. Also, in figures 11, 12, 13, 14, and 15, the influence of the number of nodes on the elapsed time to reach the optimal solution of the problem is evaluated. Firstly, it is important to realize that, in the worst-case scenario, a low elapsed time, around 0.1 seconds, was required to obtain the optimal solution. Furthermore, it is noticed that the time needed grows linearly as the number of nodes increases. This behavior may be a natural tendency of the system due to the linear formulation used, or it may be that we have not yet evaluated a significant number of nodes to be able to verify an exponential growth. Furthermore, it is possible to verify in this case that, as the average overlap increases, the time required to obtain the optimal solution to the problem also increases. Thus, the curve that evaluated an average overlap of 80% took longer to obtain the solutions than the curve with an average overlap of 20%. This is an unexpected result, because as previously mentioned, it was expected to achieve the optimal result faster with higher overlaps than with lower overlaps. And the three curves with an average overlap of 50% have a similar time for their respective numbers of nodes, as expected. It is worth pointing out that the results obtained with the classical approach are not totally comparable to the quantum approaches, as previously mentioned, due to the difference between the formulations, one is completely linear (classical), and the others are quadratic (QUBO).

As previously mentioned, to obtain these results, each combination of overlap range and number of reads was simulated 1000 times to eliminate possible computational bias. However, when checking the standard deviation of the results, it was noticed that the variation between the results was minimal for all the amounts of reads evaluated. Thus, it is possible to reduce the amount of re-sampling and decrease the total time needed to perform the simulations without compromising the result obtained.

It is recommended that analyses with a larger number of nodes be evaluated to understand how the elapsed time scales with an increasing number of reads. However, regarding the number of nodes for future studies, the current license that we have available has a limitation, in the linear case, of up to 44 nodes, with the probable need to acquire one that allows the analysis of more complex and closer to reality cases.

6.3 Quantum-inspired approach with Chaotic Amplitude Control

Using Chaotic Amplitude Control combined with random hyperparameter search, we were able to obtain optimal solutions to all problems up to 10 nodes, and one optimal solution to a 15-node problem with an overlap range of 45-55 bases. A feasible solution to the 20-node problem with an overlap of 40-60 was found. The solution is almost optimal with only one incorrect edge. The 20-node problem with an overlap of 20-80 bases was solved to feasibility, with three edged permuted compared to the optimal solution.

Larger problems could not be solved due to the solver timing out. Given a number of nodes N, the QUBO Matrix scales as O(N^4). Barring the exploitation on sparsity, since the simulation time of CAC is fixed, this gives a simple lower bound scaling of CAC in a sequential computing model as approximately employed by most modern CPUs. In practice, simulating CAC on the CPU scales significantly worse. Even without targeting a constant success probability, CAC seems to scale approximately as O(exp(N^1/2)), indicating a significant increase in difficulty in simulating the Coherent Ising Machine for large problems. Identifying the root cause would require deep investigation into the ODE solvers used, a deep understanding of the Julia programming language’s compilation process, and a tightly controlled Benchmark environment.

We do not expect these exact scaling issues to exist on a physical system, although system instability might persist. Of particular concern for CAC is the choice of hyperparameters. While the current random search within a hypercube does work for small problems, it is not efficient. Rigorous analysis of the volume of space occupied by hyperparameter sets leading to almost chaotic dynamics, which solve problems efficiently, in relation to problem and size is of great interest and would further our knowledge of CIMs considerably.

In the future, we would like to submit these problems to NTT's physical CIM, LASOLV, for evaluation.

7. CONCLUSIONS AND NEXT STEPS

We have described and evaluated the results obtained from the solvers in the quantum and classical approaches. This includes a comparison between the performance of the quantum and classical solvers. Now we present our main **conclusions** regarding the advantages and limitations of the quantum approach and possible future directions for research in this field are discussed.

- The D-Wave Hybrid solver and the classical Gurobi Linear solver have shown the best performances in obtaining optimal solutions for all the proposed scenarios.
- The purely quantum D-Wave Advantage6.1 QPU solver was only able to obtain optimal solutions for the five-node case due to scale limitations.
- The D-Wave Simulated Annealing solver obtains optimal solutions for scenarios with small number of nodes, but has difficulties in maintaining optimality as the number of nodes increases.
- Overall, the results obtained with the different solvers look promising in many aspects, but at the same time highlight the importance of choosing an appropriate solver for the problem and the need for further research to improve the scalability of purely quantum solvers.

The scope of this report and the limitations we have encountered have allowed us to think of new ideas and possible **next steps** to optimize the mathematical formulation of the model and its implementation for the solvers, to be able to address problems with higher computational capacity and to apply our model to real genomic sequencing data. Because of that, we propose possible advances and improvements to our model.

Increase the number of nodes and explore other scenarios.

We are exploring computationally larger scenarios by increasing the number of nodes. In this report, we have generated scenarios with up to 30 nodes, but what happens when we increase the number of nodes and run it on the hybrid quantum-classical devices we currently have? This can be done by optimizing the design of the model or by increasing the number of qubits. Clearly, by increasing the capacity of the hardware, we can achieve more accurate and faster genome assembly.

Graph partitioning using techniques such as Kamedias or METIS.

To solve larger graphs, we can use graph partitioning techniques that divide the graph into smaller, more manageable subgraphs. Using these techniques, we can reduce the computational burden and potentially get better results.

Alternative formulations of the model that reduce scalability issues.

To ensure the scalability of the model, we can optimize the algorithm and reduce the number of qubits required. This can be achieved by developing more efficient model designs and exploring alternative approaches to genome assembly.

Improved simulated sequences generation.

By simulating more reads and filtering them, similarly to real sequencing, we can better understand the potential of quantum computing for genome assembly. This can be achieved by incorporating more realistic error models and testing the assembly results with a larger dataset.

Handling reads with random errors.

Finally, we can explore the potential of quantum computing to handle reads with errors. To do this, we can develop algorithms capable of detecting or correcting errors in reads, which could lead to more accurate genome assembly.

8. REFERENCES

[1]. Lucas, A. Ising formulations of many NP problems. Front. Phys. 2, 5 (2014).

[2]. Cook WJ. In pursuit of the traveling salesman: mathematics at the limits of computation. Princeton University Press; 2011.

[3]. Boev, A.S., Rakitko, A.S., Usmanov, S.R. *et al.* Genome assembly using quantum and quantum-inspired annealing. *Sci Rep* **11**, 13183 (2021). https://doi.org/10.1038/s41598-021-88321-5

[4]. Sarkar A, Al-Ars Z, Bertels K (2021) QuASeR: Quantum Accelerated de novo DNA sequence reconstruction. PLoS ONE 16(4): e0249850. https://doi.org/10.1371/journal.pone.0249850

[5]. Nałęcz-Charkiewicz, K., Nowak, R.M. Algorithm for DNA sequence assembly by quantum annealing. *BMC Bioinformatics* **23**, 122 (2022). https://doi.org/10.1186/s12859-022-04661-7

[6]. Cao, Y., Romero, J., & Aspuru-Guzik, A. (2018). Potential of quantum computing for drug discovery. *IBM Jornal of Research and Development*, *62*(6), 6-1

[7]. Solenov, D., Brieler, J., & Scherrer, J. F. (2018). The potential of quantum computing and machine learning to advance clinical research and change the practice of medicine. *Missouri medicine*, *115*(5), 463.

[8]. Outeiral, C., Strahm, M., Shi, J., Morris, G. M., Benjamin, S. C., & Deane, C. M. (2021). The prospects of quantum computing in computational molecular biology. *Wiley Interdisciplinary Reviews: Computational Molecular Science*, *11*(1), e1481

[9]. Andersson, M. P., Jones, M. N., Mikkelsen, K. V., You, F., & Mansouri, S. S. (2022). Quantum computing for chemical and biomolecular product design. *Current Opinion in Chemical Engineering*, *36*, 100754.

[10] . https://www.ncbi.nlm.nih.gov/nuccore/9626372.

Contact:

Jose Aznar

Responsible for NTTDATA Health Innovation in Europe
NTT DATA Spain
jose.ignacio.aznar.baranda@nttdata.com