Genome assembly using quantum and quantum-inspired annealing

A.S. Boev,[1] A.S. Rakitko,[2] S.R. Usmanov,[1] A.N. Kobzeva,[1]
I.V. Popov,[2] V.V. Ilinsky,[2] E.O. Kiktenko,[1] and A.K. Fedorov[1]

[1] *Russian Quantum Center, Skolkovo, Moscow 143025, Russia*
[2] *Genotek ltd., Moscow 105120, Russia*

(Dated: April 23, 2020)

Recent advances in DNA sequencing open prospects to make whole-genome analysis rapid and reliable, which is promising for various applications including personalized medicine. However, existing techniques for *de novo* genome assembly, which is used for the analysis of genomic rearrangements, chromosome phasing, and reconstructing genomes without a reference, require solving tasks of high computational complexity. Here we demonstrate a method for solving genome assembly tasks with the use of quantum and quantum-inspired optimization algorithms. Within this method, we present experimental results on genome assembly using quantum annealers both for simulated data and the ϕX 174 bacteriophage. Our results pave a way for a significant increase in the efficiency of solving bioinformatics problems with the use of quantum computing technologies and, in particular, quantum annealing might be an effective method. We expect that the new generation of quantum annealing devices would outperform existing techniques for *de novo* genome assembly. To the best of our knowledge, this is the first experimental study of *de novo* genome assembly problems both for real and synthetic data on quantum annealing devices and quantum-inspired algorithms.

Over the past few decades, an amount of DNA-related data has been increasing exponentially [1], and genomics is by now a data-driven science [2]. More than 40 years ago, the first DNA genome (ϕX 174 bacteriophage) was sequenced [3]. It took almost 13 years to sequence the human genome. Today, public and private facilities offer human genome sequencing that takes days or weeks [4]. Current technologies sequence whole genomes in an unstructured set of reads with partial overlapping. However, the task of DNA assembling, which is a required step for most of the applications, is still of a challenge [5].

Existing approaches in sequencing read analysis are based on *de novo* assembling or mapping to an established reference. *De novo* assembly is essential for studying new species and structural genomic changes that cannot be detected by reading mapping. The complexity of *de novo* assembly depends on the genome size, abundance, length of repetitive sequences, and possible polyploidy. For example, *de novo* assembly of a tiny ϕX 174 genome (5386 base pairs) on a laptop takes 10 minutes, while for the human genome (3.2×10^6 base pairs) it takes about 48 hours on a supercomputer [6]. This time scale is acceptable in research but has limited use in emergency applications (including the clinical use). Read mapping on a backbone of the reference genome is computationally more simple and allows detection of single- and oligonucleotide mutations, which are the major causes of human diseases [7]. However, the detection of genome rearrangements is a challenging task [8]. Read mapping algorithms used for the analysis of clinically important samples use local *de novo* assembly to correct mapping errors and reference mismatches [9]. *De novo* assembly is currently used in transcriptome and cancer analysis, as gene fusions and genome rearrangements are common causes of malignant tumours [10]. Decreasing the costs of sequencing makes whole-genome sequencing an irreplaceable part of personalized medicine and cancer treatment. The utility of sequencing technologies requires improved workflows with *de novo* assemblers to uncover significant genomic rearrangements in cancer and normal tissues.

Early generations of assembly tools are based on the overlap layout consensus (OLC) algorithm [12]. Overlap discovery involves all-against-all, pair-wise read comparison, where one sets up the minimal number of shared nucleotides between two reads and an allowed number of mismatches. In the OLC graph, each read is represented by a vertex and an edge between two vertices indicates the overlap between corresponding reads. Thus, finding the Hamiltonian path, i.e., the path that goes through all vertices and visits each vertex only once, allows reconstructing the original genome. This approach was widely used in Sanger-era assemblers (e.g., see Ref. [13]) and becomes suitable for the single-molecule sequencing technologies like PacBio or Oxford Nanopore [14, 16].

It is well known that finding a Hamiltonian cycle belongs to the class of NP-complete computational problems, for which finding an efficient solution is very hard. That is why the other graph representation is applied to the analysis of the sequencing data. This approach is related to the concept of De Bruijn graphs (DBG) [17]. The idea is to construct a graph based on the fragmentation of reads down to smaller sequences called k-mers, where k is the length of subsequence. These k-mers are aligned using $(k-1)$ sequence overlaps. Each node in the resulting k-mer graph represents a certain k-mer, while edges correspond to the overlaps between the k-mers. Thus, each read is represented by the sequence of the connected vertices (so-called overlapped k-mers). To obtain the original genome sequence one needs to find the Eulerian path, i.e., the path that visits each edge exactly

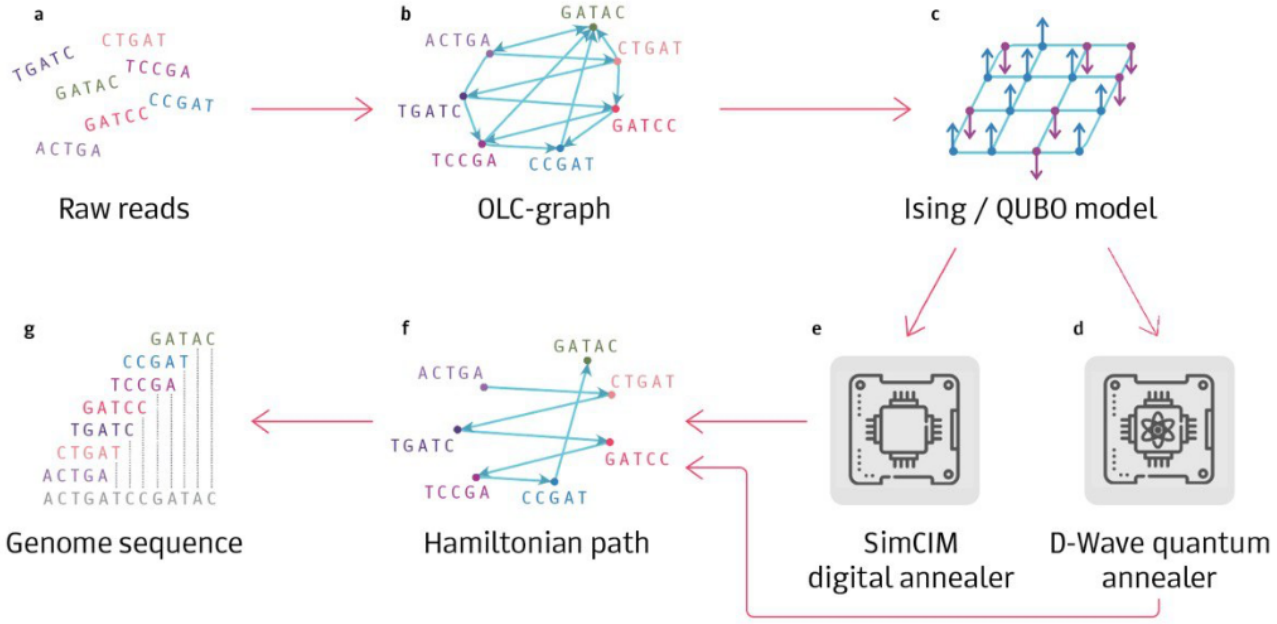

Figure 1. Solving the *de novo* genome assembly problem using quantum annealers and quantum-inspired algorithms: a) raw reads; b) raw reads are transformed to the overlap-layout-consensus (OLC) graph; c) finding the Hamiltonian path for the OLC graph is reduced to the Ising problem; d) and e) the Ising problem in QUBO form can be solved using quantum annealers (D-Wave) and quantum-inspired algorithms (SimCIM); in f) the output is the Hamiltonian path; in g) the genome sequence is obtained as the solution.

once, but is allowed to revisit any vertex. The comparison of the OLC and DBG approaches are discussed in more detail in Ref. [15]. As was mentioned above, the time of *de novo* assembling is crucial for many applications. Possible improvements and speed-ups of the DBG approach, employing the Eulerian path, have been considered, whereas methods based on finding the Hamiltonian path in the OLC approach are less studied.

Quantum computers are a new generation of computing devices that use quantum phenomena, such as superposition and entanglement, for solving computational tasks. It is widely believed that quantum computers have a great potential to outperform existing technologies in a number of hard computational tasks [18], e.g., in simulating complex systems [19], machine learning [20], and optimization [21]. A basic model of quantum computing, the so-called gate-based model [22], has been also considered as a tool for solving various tasks from the field of bioinformatics [23]. In the gate-based quantum computing framework, algorithms are constructed as a sequence of quantum logical gates (that implement various operations, such as AND, CNOT, etc.), which act on qubits (quantum analogues of logical bits that allow having a superposition of logical states 0 and 1). One of the algorithms that can be realized via gate-based quantum computing is Grover's search [24], which can be used as a subroutine for sub-sequence alignment with a quadratic speed-up [25]. There is increased activity at the interface of machine learning and quantum computing in the computational biology domain [23, 26]. Being of extreme interest from the viewpoint of obtaining polynomial and exponential computational speed-ups, the suggested methods [23–27] require both a significant number of qubits and quite low error rates. This is beyond the capabilities of existing noisy intermediate-scale quantum (NISQ) devices.

Another approach to quantum computing is known as adiabatic quantum computing [28], and it is by now well explored in solving various complex optimization tasks. The idea behind this method is to steer the dynamics of quantum many-body systems such that their final states provide solutions to optimization problems. One of the examples of available hardware for quantum annealing is the devices by D-Wave System [29–31]; however, their ability to demonstrate computation speed-up is still a subject of debates [31–36]. An interesting outcome of these debates is the appearance of a new generation of quantum-inspired algorithms, which are essentially classical but appear as a result of analysing quantum systems [37–39]. Recent results on the comparison between available quantum annealers and quantum-inspired algorithms on realistic optimization problems are of great interest [39]. We note that quantum annealing has been applied to various real-world tasks, including computational biology problems [40], exploration of the conformational landscape of peptides and proteins [41], and genome sequence alignment [27].

Here we investigate the *de novo* genome assembly

problem within the framework of quantum annealing. The main step in our study is to map the genome assembly problem in the framework of OLC graphs to a quadratic unconstrained binary optimization (QUBO) problem, which can be then efficiently embedded in the quantum annealing architecture (see Fig. 1). We also show that the genome assembly problem can be efficiently solved with the use of quantum-inspired optimization algorithms. We note that our idea is to use quantum optimization for *de novo* sequencing, while other problems related to the analysis of genetic data are beyond the scope of the present work.

QUBO REFORMULATION OF THE GENOME ASSEMBLY PROBLEM

Growing interest in adiabatic quantum optimization is related to their potential in solving combinatorial optimization (NP-hard) problems. It is widely accepted that the potential of quantum annealing is rooted in the quantum effects that allow to efficiently explore the cost-function landscape in ways unavailable to classical methods.

The physical idea behind is the adiabatic theorem of quantum mechanics, which says that a physical system will remain in the ground state if a given perturbation acts slowly enough (an additional requirement is the existence of the gap between the ground state and the rest of the energy spectrum of the system). Then one can first prepare the physical system in an initial configuration (Hamiltonian) $\mathcal{H}_0$ (not to be confused with Hamiltonian path), whose ground state is easy to obtain, and adiabatically transform it to the problem Hamiltonian $\mathcal{H}_\mathrm{P}$, whose configuration encodes the problem one is aiming to solve. Therefore, if the adiabatic conditions are fulfilled, then the ground state of the system at $t = t_a$ (where t_a is called the annealing time; the Hamiltonian of the system is $\mathcal{H}_\mathrm{P}$ at $t = t_a$) will be the solution of the optimization problem. Consequently, the important stage is to map the problem of interest to a Hamiltonian, which maps the binary representation of a graph path into a corresponding energy value.

The existing physical implementation of quantum annealing is the D-Wave quantum processor, which can be described as Ising spin Hamiltonian. The Ising Hamiltonian can be transformed into a QUBO problem. Thus, we have to find the mapping to a problem that we would like to solve on the D-Wave quantum processor to the QUBO form. In this case, each spin (qubit) represents a variable, and couplers between qubits represent the costs associated with qubit pairs (see Methods).

Hamiltonian path mapping

Along the lines of Ref. [42], we reformulate the task of finding the longest Hamiltonian path in the OLC graph as a QUBO problem. We assume that the OLC graph is defined as $G = (V, E)$, $V = |N|$, where we label the vertices as $1, \ldots, N$ and the edge set as (u, v). Our solution uses N^2 binary variables $x_{v,i}$ that are associate with spins (qubits), where v represents the vertex and i represents its order in a prospective path $(v, i \in 1, \ldots, N)$. We note that it has been theoretically shown that fewer bits can be used for such representation, but we focus on the most simple form. These requirements are encoded in the Hamiltonian as follows:

$$\begin{aligned}\mathcal{H} = A\sum_{v=1}^{N}\left(1-\sum_{j=1}^{N}x_{v,j}\right)^2 \\ + A\sum_{j=1}^{N}\left(1-\sum_{v=1}^{N}x_{v,j}\right)^2 \\ + A\sum_{(u,v)\notin E}\sum_{j=1}^{N-1}(x_{u,j}x_{v,j+1}),\end{aligned} \tag{1}$$

where A is the normalization coefficient. The energy then has three components. The first two conditions require that every vertex can only appear once in a path, and that there must be a j-th node in the path for each j. Finally, for the nodes in prospective ordering, i.e., if $x_{u,j}$ and $x_{v,j+1}$ are both 1, then there should be an energy penalty if $(u, v) \notin E$. With this QUBO formulation we are able to run the genome assembly task using quantum annealers and quantum-inspired algorithms.

We note that the applicability of the method requires the existence of the Hamiltonian path in the corresponding graph, which is not universally the case for arbitrary genetic data. It is also important to note that a polynomial overhead in the number of spins is required.

Hamiltonian path transformation (For acyclic graphs)

In general, Hamiltonian path mapping is suitable both for cyclic and acyclic graphs. However, it is often the case that the OLC graph contains no cycles. It is then possible to further simplify transformation and reduce the qubit overhead. Here, we demonstrate more compact mapping that requires only N qubits, where N is the number of edges. For the same OLC graph $G = (V, E)$ and the edge set (u, v) let us define binary variable $x_{u,v}$ that indicates whether the edge (u, v) is included in path. Then the corresponding Hamiltonian should include the following

two components:

$$\mathcal{H} = \sum_{v \in V} \left(1 - \sum_{(u,v) \in E} x_{u,v}\right)^2 + \sum_{v \in V} \left(1 - \sum_{(v,u) \in E} x_{v,u}\right)^2 \quad (2)$$

The first component requires that each vertex is incident with a single incoming path edge. The second component requires that each vertex is incident with a single outgoing path edge. Although this realization is helpful and can be used for solving genome assembly problems on quantum annealers without polynomial qubit overhead, the asymptotic computational speed-up versus classical algorithm is not exponential.

RESULTS

We now proceed with the results of our experimental realization of *de novo* genome assembly using quantum and quantum-inspired annealers. We start with the paradigmatic example of the ϕX 174 bacteriophage genome. In order to realize *de novo* genome assembly, we construct the adjacency matrix for OLC graphs and use pre-processing for packing this graph into D-Wave processor (see Methods). We then transform each adjacency graph into the QUBO matrix according to Eq. (1). In this form the problem can be solved with the use of quantum annealing hardware by D-Wave and quantum-inspired optimization algorithm. For each instance, a total of 10^3 anneals (runs) were collected from the processor, with each run having an annealing time of 20 μs. Similar routine was realized with the use of quantum-inspired annealing. Up to our best knowledge, this is the first realistic-size *de novo* genome assembly employing the use of quantum computing devices and quantum-inspired algorithms. The results are presented in Table I.

Benchmarking quantum-assisted *de novo* genome assembly

In order to perform a complete analysis of the suggested approach, we realize the quantum-assisted *de novo* genome assembly for synthetic dataset. We generate a synthetic dataset, which consists of 60 random reads of length from 5 to 10 (for details, see Methods). We then split each read into k-mers of length 3 and compute adjacency matrix for the corresponding OLC graph. Finally, we transform each adjacency graph into the QUBO matrix according to our algorithm and minimize it using quantum annealing hardware by D-Wave and quantum-inspired optimization algorithms.

Our goal is to check the applicability of existing quantum annealers to the task of genome assembly, evaluate the upper bound on the input problem size (particularly, the length of the original genome), compare the performance of the D-Wave quantum annealer with a software annealing simulator SimCIM. The choice of tools is motivated by their maturity n terms of quantum dimensionality and compatibility with the original formulation in terms of the optimization problem.

	Mean, ms	Min, ms	Max, ms	90% Percentile
Quantum annealer (D-Wave)	9018	8985	9063	9055
Quantum-inspired annealer (SimCIM)	262	9.9	7212	1061

Table I. Genome assembly time for ϕX 174 bacteriophage.

As a figure of merit we use the success probability (SP) and time-to-solution (TTS). We define the success probability as a ratio of annealing runs that converged to the global minimum (true ground state) and reconstructed the original genome sequences, to the total number of runs as follows:

$$\mathrm{SP} = \frac{n_{\text{anneal runs}}(\text{ground state})}{n_{\text{anneal runs}}(\text{total})}, \quad (3)$$

In turn, TTS is inversely proportional to the success probability and shows the total computing time necessary to reconstruct genome sequence with high certainty. We define it in the following way:

$$\mathrm{TTS} = t_a/\mathrm{SP}. \quad (4)$$

where t_a for D-Wave is $20\mu s$ (default value).

Quantum-inspired optimization algorithms can be also used for solving QUBO problems. In our experiments, we employ SimCIM quantum-inspired optimization algorithm [37], which is based on the differential approach to simulating specific quantum processors called Coherent Ising Machine (CIM; see Methods). SimCIM runs on conventional hardware and is easily parallelizable on graphical processing units (GPU). This is the time for simulating a single annealing run using our implementation of SimCIM, measured on Intel core i7-6700 Quad-Core, 64GB DDR4, GeForce GTX 1080. We test the suggested approach with the simulated data first with the D-Wave quantum annealer (see Fig. 2) and compare our results with quantum-inspired optimization algorithm SimCIM. D-Wave shows an advantage in genome assembly for short-length sequences, while it cannot be applied for sequences of length 7 and more due to the fact that the decoherence time becomes comparable with the annealing time.

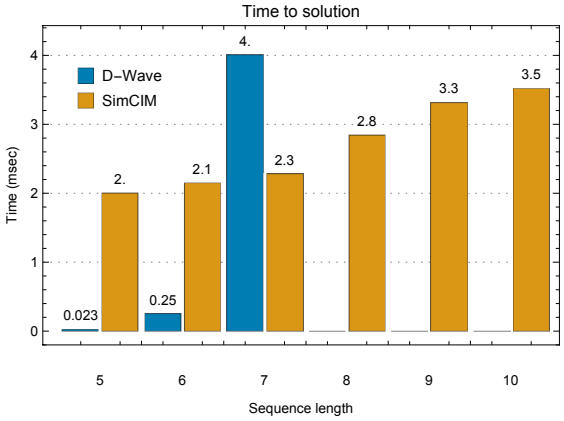

Figure 2. Comparison of the performance of quantum and quantum-inspired methods for *de novo* genome assembly based on synthetic data: we compare quantum device D-Wave and quantum-inspired optimization algorithm SimCIM.

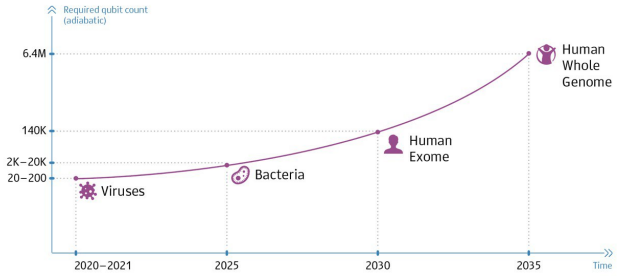

Figure 3. Forecasts on the use of quantum computing devices for solving the genome assembly task.

Forecasts

Limitations of existing quantum hardware do not allow universally outperform existing solutions for *de novo* genome assembling. At the same time, one of the most interesting practical questions is when one can expect computational advantages from the use of quantum computing in genome assembling tasks.

We obtain a rough time estimate for the applicability of quantum genome assembly for various genome lengths by extrapolating the qubit count using Moore's Law (double every year from 2,000 qubits in 2017), and then comparing the extrapolated value with the necessary volume, the latter depending on the taxonomic kingdom. The required number of qubits can be roughly estimated as the ratio of the genome size length to the single read length, which is the estimate for the QUBO matrix size (see Fig. 3).

DISCUSSIONS

In our work, we have demonstrated the possibility of solving the simplified bioinformatics problem of reconstructing genome sequences using quantum annealing hardware and quantum-inspired algorithms. We have implemented experimental quantum-assisted assembly of ϕX 174 bacteriophage genome. On the basis of synthetic data, we have shown that the existing quantum hardware allows reconstructing short sequences of up to 7 nucleotides. In order to use quantum optimization for realistic tasks, the ratio of the decoherence time to the annealing time should be considerably improved. We note that while the decoherence time is not a fundamental limitation of the technology, realization of quantum annealers with sufficient decoherence time remains a challenge. While D-Wave machines use superconducting quantum circuits [43], setups based on ultracold Rydberg atom arrays [43] and trapped ions [44] can be also used for the efficient implementation of quantum annealing and other quantum optimization algorithms. Specifically, the system of Rydberg atom arrays has been studied in the context of solving the maximum independent set problem [45], which is NP-hard. For longer sequences, as we have demonstrated, it is possible to use quantum-inspired algorithms that are capable of solving more complex problems using classical hardware.

We note that our work is a proof-of-principle demonstration of the possibility to use existing quantum devices for solving the genome assembly problem. The problem scale considered in this paper is still far from real sequences (∼130 kilo-base pairs for primitive bacterias) and is lacking numerous complications, such as errors in sequence reads and handling repeating sequences. However, the proposed method demonstrates that newly evolving computing techniques based on quantum computers and quantum-inspired algorithms is quickly developing and can be soon applied in new areas of science.

We note that in real-life conditions a number of additional challenges arise. Examples include errors (random insertions and deletions, repeats, etc.), genome contaminants (pieces of the genome not related to the subject of interest), polymer chain reaction artefacts, and others require additional post-processing steps. These problems are beyond the scope of our proof-of-principle demonstration and they should be considered in future. Another complication comes from the fact that temperature and other noise effects play a significant role in the case of the use of realistic quantum devices. Thermal excitation and relaxation processes affect performance. Our further directions include optimization of QUBO model for more compact spin representation and integration of error model into our algorithm. Solving these two issues can enable reconstruction of real sequences using the quantum approach.

Our source code for a proof-of-principle realization of the quantum-assisted genome assembly is freely available under the GNU general public license. The realization also contains the synthetic set that can be used for the reproducing the obtained results. For the further im-

plementation we refer the reader to the description of D-Wave's Software.

ACKNOWLEDGEMENTS

We are grateful to A.I. Lvovsky for fruitful discussions as well as A.S. Mastiukova and D.V. Kurlov for useful comments. We thank E.S. Tiunov for providing information about the SimCIM algorithm and A.E. Ulanov for the discussion of various quantum-inspired optimization algorithms. This work is supported by Russian Science Foundation (19-71-10092).

Note. Recently, we became aware of the work reporting studies of the quantum acceleration using gate-based and annealing-based quantum computing [46].

[1] Z.D. Stephens, S.Y. Lee, F. Faghri, R.H. Campbell, C. Zhai, M.J. Efron, R. Iyer, M.C. Schatz, S. Sinha, and G.E. Robinson, Big data: astronomical or genomical? PLoS Biol. **13**, e1002195 (2015).

[2] G. Eraslan, Z. Avsec, J. Gagneur, and F.J. Theis, Deep learning: new computational modelling techniques for genomics, Nat. Rev. Genet. **20**, 389 (2019).

[3] F. Sanger, G.M. Air, B.G. Barrell, N.L. Brown, A.R. Coulson, J.C. Fiddes, C.A. Hutchison III, P.M. Slocombe, and M. Smith, Nucleotide sequence of bacteriophage φX174 DNA, Nature **265**, 687 (1997).

[4] S. Park and J. Kim, Trends in next-generation sequencing and a new era for whole genome sequencing, Int. Neurourol. J. **20**, 76 (2016).

[5] X. Liao, M. Li, Y. Zou, F.-X. Wu, Yi-Pan, and J. Wang, Current challenges and solutions of *de novo* assembly, Quant. Biol. **7**, 90 (2019).

[6] K. Wong, M. Levy-Sakin, and P. Kwok, *De novo* human genome assemblies reveal spectrum of alternative haplotypes in diverse populations, Nat. Commun. **9**, 3040 (2018).

[7] H. Lee, K. Lee, T. Lee, D. Park, J. Chung, C. Lee, W.-Y. Park, and D.-S. Son, Performance evaluation method for read mapping tool in clinical panel sequencing, Genes Genomics. **40**, 189 (2018).

[8] R. Yao, T. Yu, Y. Qing, J. Wang, and Y. Shen, Evaluation of copy number variant detection from panel-based next-generation sequencing data, Mol. Genet. Genomic. Med. **7**, e00513 (2019).

[9] Y. Li, D. Xue, B. Zhang, and J. Liu, An optimized approach for local *de novo* assembly of overlapping paired-end RAD reads from multiple individuals, R Soc Open Sci. **5**, 171589 (2018).

[10] V. Mittal and J. McDonald, *De novo* assembly and characterization of breast cancer transcriptomes identifies large numbers of novel fusion-gene transcripts of potential functional significance, BMC Med. Genomics **10**, 53 (2017).

[11] J.R. Miller, S. Koren, and G. Sutton, Assembly algorithms for next-generation sequencing data, Genomics **95**, 315 (2010).

[12] E.W. Myers, The fragment assembly string graph, Genomics **21**, ii79 (2005).

[13] E. Myers et al., A whole-genome assembly of Drosophila, Science **287**, 2196 (2000).

[14] H. Li, Minimap and miniasm: Fast mapping and de novo assembly for noisy long sequences, Bioinformatics **32**(14), 2103 (2016).

[15] Z. Li, Y. Chen, D. Mu, J. Yuan, Y. Shi, H. Zhang, J. Gan, N. Li, X. Hu, B. Liu, B. Yang, and W. Fan, Comparison of the two major classes of assembly algorithms: overlap-layout-consensus and de-bruijn-graph, Briefings in Functional Genomics **11**(1), 25 (2012).

[16] S. Koren, B. Walenz, K. Berlin, J. Miller, N. Bergman, and A. Phillippy, Canu: scalable and accurate long-read assembly via adaptive k-mer weighting and repeat separation, Genome research **27**(5), 722 (2017).

[17] P.E.C. Compeau, P.A. Pevzner, and G. Tesler, How to apply de Bruijn graphs to genome assembly, Nat. Biotechnol. **29**, 987 (2011).

[18] A.W. Harrow and A. Montanaro, Quantum computational supremacy, Nature **549**, 203 (2017).

[19] S. Lloyd, Universal quantum simulators, Science **273**, 1073 (1996).

[20] J. Biamonte, P. Wittek, N. Pancotti, P. Rebentrost, N. Wiebe, and S. Lloyd, Quantum machine learning, Nature (London) **549**, 195 (2017).

[21] E. Farhi and A.W. Harrow, Quantum supremacy through the quantum approximate optimization algorithm, arXiv.org:1602.07674.

[22] T.D. Ladd, F. Jelezko, R. Laflamme, Y. Nakamura, C. Monroe, and J.L. O'Brien, Quantum computers, Nature (London) **464**, 45 (2010).

[23] P.S. Emani, J. Warrell, A. Anticevic, S. Bekiranov, M. Gandal, M.J. McConnell, G. Sapiro, A. Aspuru-Guzik, J. Baker, M. Bastiani, P. McClure, J. Murray, S.N. Sotiropoulos, J. Taylor, G. Senthil, T. Lehner, M.B. Gerstein, and A.W. Harrow, Quantum computing at the frontiers of biological sciences, arXiv.org:1911.07127.

[24] L.K. Grover, A fast quantum mechanical algorithm for database search, in *Proceedings of 28th Annual ACM Symposium on the Theory of Computing (New York, USA, 1996)*, p. 212.

[25] A. Sarkar, Z. Al-Ars, C.G. Almudever, and K. Bertels, An algorithm for DNA read alignment on quantum accelerators, arXiv.org:1909.05563.

[26] K. Prousalis and N. Konofaos, A quantum pattern recognition method for improving pairwise sequence alignment, Sci. Rep. **9**, 7226 (2019).

[27] O.B. Lindvall, Quantum methods for sequence alignment and metagenomics (2019), PhD thesis.

[28] T. Albash and D. A. Lidar, Adiabatic quantum computation, Rev. Mod. Phys. **90**, 015002 (2018).

[29] S. Boixo, T. Albash, E.M. Spedalieri, N. Chancellor, and D.A. Lidar, Experimental signature of programmable quantum annealing, Nature Comm. **4**, 3067 (2013).

[30] S. Boixo, T.F. Rønnow, S.V. Isakov, Z. Wang, D. Wecker, D.A. Lidar, J.M. Martinis, and M. Troyer, Evidence for quantum annealing with more than one hundred qubits, Nat. Phys. **10**, 218 (2014).

[31] T.F. Rønnow, Z. Wang, J. Job, S. Boixo, S.V. Isakov, D. Wecker, J.M. Martinis, D.A. Lidar, and M. Troyer, Defining and detecting quantum speedup, Science **345**, 420 (2014).

[32] S. Woo Shin, G. Smith, J.A. Smolin, and U. Vazirani, How "quantum" is the D-Wave machine? arXiv.org:1401.7087.
[33] H.G. Katzgraber, F. Hamze, and R.S. Andrist, Glassy chimeras could be blind to quantum speedup: Designing better benchmarks for quantum annealing machines, Phys. Rev. X **4**, 021008 (2015).
[34] D. Venturelli, S. Mandrà , S. Knysh, B. O'Gorman, R. Biswas, and V. Smelyanskiy, Quantum optimization of fully connected spin glasses, Phys. Rev. X **5**, 031040 (2015).
[35] I. Hen, J. Job, T. Albash , T.F. Rønnow, M. Troyer, and D.A. Lidar, Probing for quantum speedup in spin-glass problems with planted solutions, Phys. Rev. A **92**, 042325 (2015).
[36] M.H. Amin, Searching for quantum speedup in quasistatic quantum annealers, Phys. Rev. A **92**, 052323 (2015).
[37] E.S. Tiunov, A.E. Ulanov, and A.I. Lvovsky, Annealing by simulating the coherent Ising machine, Opt. Exp. **27**, 10288 (2019).
[38] K.P. Kalinin and N.G. Berloff, Global optimization of spin Hamiltonians with gain-dissipative systems, Sci. Rep. **8**, 17791 (2018).
[39] J.M. Arrazola, A. Delgado, B.R. Bardhan, and S. Lloyd, Quantum-inspired algorithms in practice, arXiv.org:1905.10415.
[40] R.Y. Li, R. Di Felice, R. Rohs, and D.A. Lidar, Quantum annealing versus classical machine learning applied to a simplified computational biology problem, npj Quantum Inf. **4**, 14 (2018).
[41] A. Perdomo-Ortiz, N. Dickson, M. Drew-Brook, G. Rosem, and A. Aspuru-Guzik, Finding low-energy conformations of lattice protein models by quantum annealing, Sci. Rep. **2**, 571 (2012).
[42] A. Lucas, Ising formulations of many NP problems, Front. Physics. **2**, 5 (2014).
[43] H. Bernien, S. Schwartz, A. Keesling, H. Levine, A. Omran, H. Pichler, S. Choi, A.S. Zibrov, M. Endres, M. Greiner, V. Vuletić, and M.D. Lukin, Probing many-body dynamics on a 51-atom quantum simulator, Nature (London) **551**, 579 (2017).
[44] J. Zhang, G. Pagano, P. W. Hess, A. Kyprianidis, P. Becker, H. Kaplan, A.V. Gorshkov, Z.-X. Gong, and C. Monroe, Observation of a many-body dynamical phase transition with a 53-qubit quantum simulator, Nature (London) **551**, 601 (2017).
[45] H. Pichler, S.-T. Wang, L. Zhou, S. Choi, and M.D. Lukin, Quantum optimization for maximum independent set using Rydberg atom arrays, arXiv.org:1808.10816.
[46] A. Sarkar, Z. Al-Ars, and K. Bertels, QuASeR: Quantum accelerated *de novo* DNA sequence reconstruction, arXiv.org:2004.05078.
[47] F. Angly, D. Willner, F. Rohwer, P. Hugenholtz, G.Tyson, Grinder: a versatile amplicon and shotgun sequence simulator, Nucleic Acids Res. **20**(12), e94 (2012).
[48] H. Li, Minimap2: pairwise alignment for nucleotide sequences, Bioinformatics **34**(18), 3094 (2018).

METHODS

Quantum annealing

Adiabatic quantum optimization relies on the following process: Suppose we have a quantum systems with a Hamiltonian (the energy operator) $\mathcal{H}_{\rm P}$, whose ground state encodes the solution to a problem of interest, and another easy-to-prepare quantum system with another (beginning) Hamiltonian $\mathcal{H}_0$. Then one can prepare a quantum system to be in the ground state of the Hamiltonian $\mathcal{H}_0$ and adiabatically change the configuration of the system is as follows [42]:

$$\mathcal{H}(t) = \left(1 - \frac{t}{t_a}\right)\mathcal{H}_0 + \frac{t}{t_a}\mathcal{H}_{\rm P} \tag{5}$$

The adiabatic theorem says that a physical system will remain in the ground state if a given perturbation acts slowly enough and if there is a gap between the ground state and the rest of the energy spectrum of the system. Thus, at time T by measuring the quantum state a solution of the problem of interest can be obtained if all required conditions are fluffed.

The beginning Hamiltonian of the D-Wave processor is a transverse magnetic field of the following form:

$$\mathcal{H}_0 = \sum_{i\in V} h_i \sigma_i^x, \tag{6}$$

where σ_i^x is the Pauli x-matrix, which acts on ith qubit. The problem Hamiltonian can be encoded to the following Ising Hamiltonian:

$$\mathcal{H}_{\rm P} = \sum_{i\in V} h_i \sigma_i^z + \sum_{(i,j)\in E} J_{ij}\sigma_i^z\sigma_j^z, \tag{7}$$

where h_i describe local fields, J_{ij} stands for couplings, σ_i^z are the Pauli z-matrices, and E is the set of edges. One can see that $\mathcal{H}_{\rm P}$ is of diagonal form, so σ_i^z can be treated as spin values $\{\sigma_i^z = \pm 1\}$. For a given spin configuration σ_i^z the total energy of the system is given by $\mathcal{H}_{\rm P}$, so by measuring the energy one can find a solution to the problem of interest.

Quantum annealing can be applied to any optimization problem that can be expressed in the QUBO form. The idea is then to reduce the problem of interest to the QUBO form.

QUBO transformation

The Ising Hamiltonian can be directly transformed to a quadratic unconstrained binary optimization (QUBO) problem. The following transformation can be applied for this purpose:

$$w_i = \frac{\sigma_i^z + 1}{2} \in \{0, 1\}, \tag{8}$$

where $\{\sigma_i^z = \pm 1\}$. For solving the problem on the D-Wave quantum processor, all h_i and J_{ij} values are scaled to lie between -1 and 1. As a result, the processor outputs a set of spin values $\{\sigma_i^z = \pm 1\}$ that attempts to minimize the energy, and the lower energy indicates better solution of the optimization problem. We note that Ref. [42] provides a method for QUBO/Ising formulations of many NP problems.

Quantum-inspired annealing using SimCIM

SimCIM is an example of a quantum-inspired annealing algorithm, which works in an iterative manner. It can be used for sampling low-energy spin configurations in the classical Ising model. The algorithm treats each spin value s_i as a continuous variable, which lie in the range $[-1, 1]$. Each iteration of the SimCIM algorithm starts with calculating the mean field

$$\Phi_i = \sum_{j \neq i} J_{ij} s_j + b_i, \tag{9}$$

which act on each spin by all other spins (b_i is an element of the bias vector). Then the gradients for the spin values are calculated according to $\Delta s_i = p_t s_i + \zeta \Phi_i + N(0, \sigma)$, where p_t, ζ are the annealing control parameters and $N(0, \sigma)$ is the noise of the Gaussian form. Then the spin values are updated according to $s_i \leftarrow \phi(s_i + \Delta s_i)$, where $\phi(x)$ is the activation function

$$\phi(x) = \begin{cases} x \text{ for } |x| \leq 1; \\ x/|x| \text{ otherwise} \end{cases} \tag{10}$$

After multiple updates, the spins will tend to either -1 or $+1$ and the final discrete spin configuration is obtained by taking the sign of each s_i.

Bacteriophage simulations

We use *Grinder* [47] to simulate raw reads from ϕX 174 bacteriophage complete genome (NCBI Reference Sequence: NC__001422.1). To simplify the task and make it feasible for quantum computing we generate 50 reads in each run of simulations. In our proof-of-concept research, we are focused on finding the Hamiltonian path in OLC graph and the questions about building this graph is out of our interests. We generate the raw reads with no sequencing errors and the length of each read is equal to 600 base pairs. The average coverage in this mode is about $5.7x$. We build the OLC graph using the pairwise alignment of the raw reads implemented in *minimap2* package [48]. We run *minimap2* with the predefined set of parameters ava-ont and $k = 10$. We apply *miniasm* [14] to the same data as the benchmark assembler, which uses

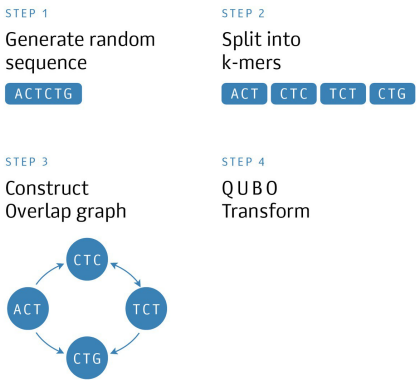

Figure 4. Experimental scheme for the synthetic dataset.

OLC graphs. Finally, we run quantum annealing hardware by D-Wave and quantum-inspired optimization algorithm for solving the assembly problem.

For experiments with quantum annealing, we use public access to D-Wave 2000Q via Leap SDK. We evaluate the impact of tunable parameters (particularly, annealing time) on the final solution quality; however, no significant improvement was discovered against default values, so annealing time was set to 20 μs (default value). The number of annealing runs is set to 10,000 (maximum possible value). Synthetic dataset graphs up to the length of 7 nucleotides (25 graph nodes) are small enough to fit into quantum annealer, so we can use DW__2000Q__5 backend (pure quantum mode of operation; see the following section). However, due to the large size of the ϕ-X 174 bacteriophage graph (248 vertices), we use the heuristic graph decomposition technique (using Metis tool) and split the graph into 3 parts of similar size. While each part is still large to fit completely into the D-Wave quantum processors, we use newly released hybrid mode of computation — hybrid__v1 backend. According to D-Wave Leap specification, hybrid__v1 backend automatically combines the power of classical and quantum computation.

Simulations with synthetic dataset

In order to evaluate the performance of the algorithm in a controlled setup, we generated several hundreds of random nucleotide sequences with variable length and performed corresponding transformations as shown in Fig. 4. Further, we eliminated graph duplicates or other trivial cases, where graph structure contained no auxiliary edges. Synthetic dataset graphs up to the length of 7 nucleotides (25 graph nodes) are small enough to fit into quantum annealer, so we can use DW__2000Q__5 backend (pure quantum mode of operation). Finally, we selected 60 sequences that produce unique OLC graphs with comparable complexity.