Molecular Biotechnology (2024) 66:163–178
https://doi.org/10.1007/s12033-023-00765-4

REVIEW

Quantum Computing in the Next-Generation Computational Biology Landscape: From Protein Folding to Molecular Dynamics

Soumen Pal[1] · Manojit Bhattacharya[2] · Sang-Soo Lee[3] · Chiranjib Chakraborty[4]

Received: 14 March 2023 / Accepted: 4 May 2023 / Published online: 27 May 2023

Abstract
Modern biological science is trying to solve the fundamental complex problems of molecular biology, which include protein folding, drug discovery, simulation of macromolecular structure, genome assembly, and many more. Currently, quantum computing (QC), a rapidly emerging technology exploiting quantum mechanical phenomena, has developed to address current significant physical, chemical, biological issues, and complex questions. The present review discusses quantum computing technology and its status in solving molecular biology problems, especially in the next-generation computational biology scenario. First, the article explained the basic concept of quantum computing, the functioning of quantum systems where information is stored as qubits, and data storage capacity using quantum gates. Second, the review discussed quantum computing components, such as quantum hardware, quantum processors, and quantum annealing. At the same time, article also discussed quantum algorithms, such as the grover search algorithm and discrete and factorization algorithms. Furthermore, the article discussed the different applications of quantum computing to understand the next-generation biological problems, such as simulation and modeling of biological macromolecules, computational biology problems, data analysis in bioinformatics, protein folding, molecular biology problems, modeling of gene regulatory networks, drug discovery and development, mechano-biology, and RNA folding. Finally, the article represented different probable prospects of quantum computing in molecular biology.

Keywords Quantum computing · Molecular biology · Simulation · Computational biology

Introduction

Modern biological science is trying to solve the fundamental complex problems of biology, incredibly in the domain of structural and functional biology. Several areas of modern biological problems include protein folding, drug discovery, simulation of macromolecular structural mobility, genome assembly, and many more. To solve the critical questions in modern biological science, scientists are solving the problems using computational biology, system biology, statistical and mathematical models, and computational algorithms. Researchers require advanced computation power and next-generation experimental methods to demonstrate more complex biological phenomena. In recent times, many real-life problems in biology have become more challenging as they need a significant quantity of computational resources. Therefore, in the same space, speedy supercomputing and enormous parallel computing facilities are the need of the present day, which is changing the present scenario of computational resources and creating entirely new computing paradigms. Now, quantum computing (QC) has developed a plethora of ways to address modern biological problems and complex biological questions [1–4].

The promise of quantum computing has started with the execution of the faster on a quantum processor. In the year

Soumen Pal, Manojit Bhattacharya and Chiranjib Chakraborty have contributed equally to this work.

✉ Chiranjib Chakraborty
drchiranjib@yahoo.com

1 School of Mechanical Engineering, Vellore Institute of Technology, Vellore, Tamil Nadu 632014, India

2 Department of Zoology, Fakir Mohan University, Vyasa Vihar, Balasore, Odisha 756020, India

3 Institute for Skeletal Aging & Orthopedic Surgery, Hallym University-Chuncheon Sacred Heart Hospital, Chuncheon, Gangwon-Do 24252, Republic of Korea

4 Department of Biotechnology, School of Life Science and Biotechnology, Adamas University, Kolkata, West Bengal 700126, India

of 2019, researchers executed quantum supremacy successfully in a 53 qubit superconducting machine. In this study, researchers have used a programmable superconducting qubit processor and generated quantum states on 53 qubits. With this experiment, the possibility of the fast computing using quantum computing has been unlocked with the new computational capacity. The researchers have explained that their sycamore processor takes about 200s to compute. A similar job would take approximately 10,000 years by a state-of-the-art classical supercomputer [5]. After that, Zhong et al. generated up to 76 output photon clicks. This experiment yielded an output state-space dimension of 10^{30} by the photonic quantum computer, Jiuzhang. It is the fastest calculation than the state-of-the-art simulation approach [6].

Planck, Bohr, and Einstein were the founders of quantum theory in the early twentieth century; Feynman and Manin introduced a revolutionary idea in the 1980s, i.e., quantum computing. The underlying rule postulated in quantum theory was that energy can be considered as distinct packets termed 'quantum,' and this empowered researchers to express energy–matter interactions in the realm of subatomic understanding, in quantum computing. In 1980, the first mechanical model of a computer was developed using the Schrodinger equation. It was known as Quantum Turing Machine (QTM). Paul Benioff illustrated the first QTM model of a quantum computer. After that, the quantum computer dramatically evolved from time to time to reach the present-day quantum computer architecture based on trapped-ion QCCD (Fig. 1). However, in quantum computing, the fundamental essence is to keep information at the quantum level of matter and apply quantum gate action to evaluate that information by programming quantum interference. Then, coming to the scope of what quantum computing can achieve in biological interactions, its contribution is possible in biochemical systems [7], biology [8], biochemistry [9], computation of protein–ligand interplay [10], mRNA codon optimization [11], and protein folding [12] to name a few with higher-order speeds compared with conventional computers. To understand this higher velocity magnitude, we must accept that quantum computers work differently than their traditional counterparts. It is pertinent to mention that using the quantum processor for processing quantum information requires a rudimentary switch in comprehending 'programming' as we do today. However, this article presents the quantum information principles and in what way these are utilized to undertake computation. The article elucidates the functioning of quantum systems where information is stored as qubits and in what capacity this data could be used with quantum gates. In quantum computing, the elementary constituents of algorithms comprise qubits and quantum gates. Further, the technical hurdles in quantum computer development are also highlighted. Finally, the review represented a wide range of probable applications of molecular biology in future days.

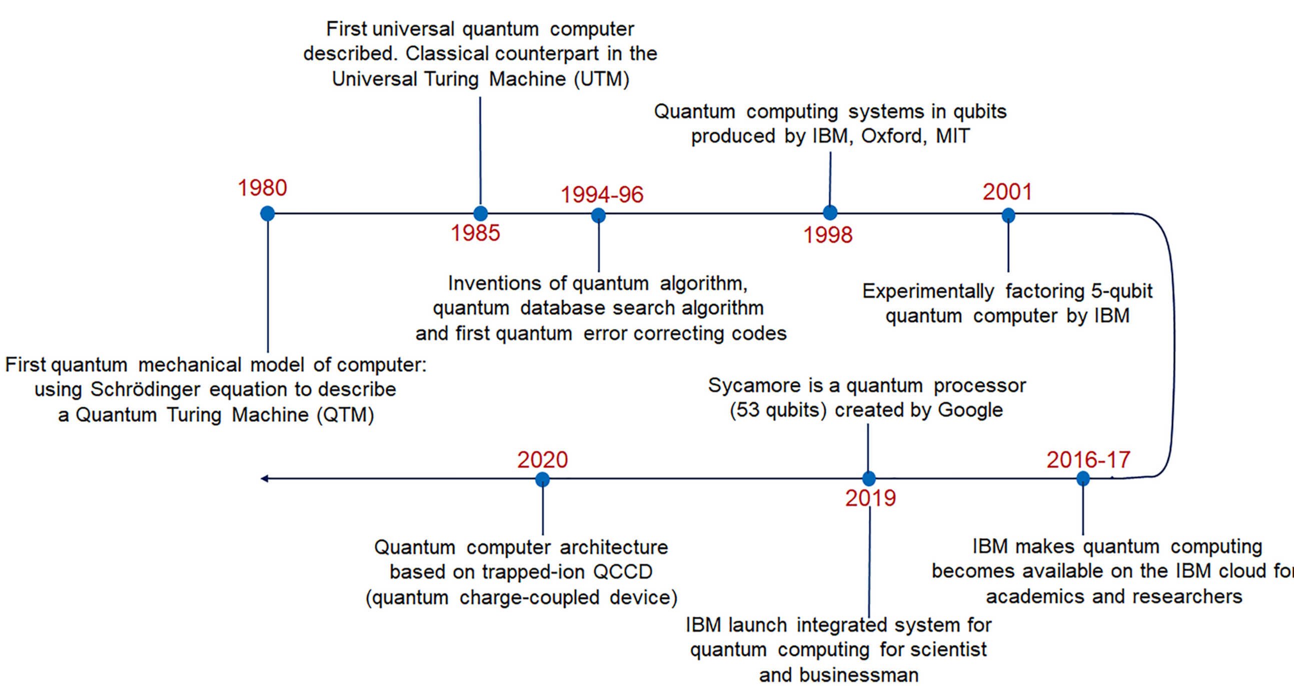

Fig. 1 Evolving path of a quantum computer with a timeline. The evolving pathway shows, from Quantum Turing Machine (QTM), how quantum computers dramatically evolved from time to time to reach the present-day quantum computer architecture based on trapped-ion QCCD

Quantum Computing Using Quantum Mechanics: New Notations

Quantum computing has used several new notations. We have provided several notations, auxiliary qubit, NOT gate, identity gate, quantum assembly language (QASM), qiskit program, qubit, toffoli gate, hadamard (H) gate, Y gate, phase gate, classical gate, etc. in Supplementary Table S1 following Dirac notation [13]. Several standard textbooks have used the notations [14, 15]. Some examples are QASM, toffoli gate, hadamard (H) gate, etc. QASM is a set of text-based instructions to describe and visualize quantum circuits. Similarly, the toffoli gate is a double-controlled-NOT gate (CCX) with two control qubits and one target. Likewise, the hadamard gate rotates the states $|0>$ and $|1>$ to $|+>$ and $|->$, one to one. It helps to make the superpositions. Some significant notations are as follows:

Qubit: Quantum Bit and Quantum Information

Quantum bit, i.e., qubit in quantum computing, is the elementary component of information and can be considered analogous to the 'bit' in classical computing (Fig. 2). The quantum processor, which is used for quantum computing, handles information in the form of qubits. Qubits can be considered conceptual mathematical entities [15]. The use of qubits as conceptual objects is of great, advantage because quantum computation theory can be built for quantum data processing independent of any particular system. A classical bit can be associated with states 0 and 1. In classical computers, binary numbers are represented by only two symbols or digits, 0 and 1. Likewise, a qubit also has a state. However, along with 0 and 1, it can adopt any combination of the states 0 and 1. Interestingly, when viewed, the superposition no longer exists, and the qubit collapses to either 0 or 1, which is analogous to Schrodinger's cat being living or dead [16]. Another hallmark feature of the qubit is that when several are amalgamated together, they probably get correlated, and interactivity with any one unit of these has fine-drawn intimation in the whole system state [17]. It is well established that this interconnection among several qubits, termed 'quantum entanglement,' is essential to perform quantum computing.

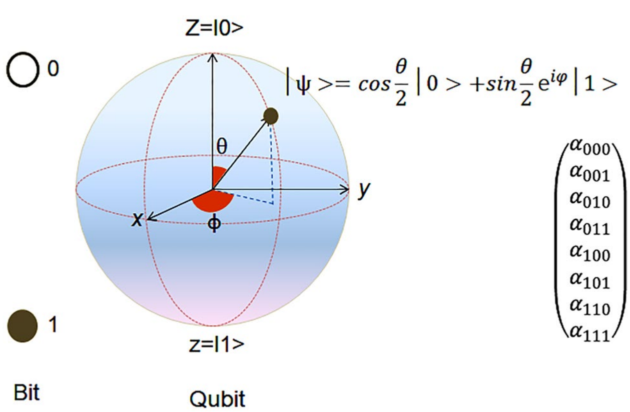

Fig. 2 The figure depicts the conceptual design of bit and qubit. A qubit state represents a unit of a sphere with the north and south poles. The north and south poles correspond to the states 0 and 1 of a classical bit. The figure also illustrates that the space of 3 qubits is a 2^3-dimensional complex vector

A qubit can be expressed in mathematical form with the idea that the quantum analog of a bit is referred to as a qubit. The two states of the qubit are denoted as $|0\rangle$ and $|1\rangle$ which one may assume to link these with 0 and 1 states of bit. Here, Dirac notation is used and $|\cdot\rangle$ is represented as quantum state [13]. The main distinction qubit displays, when compared with bits, is that it may exist in states in addition to $|0\rangle$ or $|1\rangle$. This phenomenon is known as the superposition of $|0\rangle$ and $|1\rangle$ states and is expressed as follows [17]:

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle \quad \alpha, \beta \in \mathbb{C} \quad |\alpha|^2 + |\beta|^2 = 1 \tag{1}$$

where α and β are coefficients having complex form and designated as amplitudes of the states $|0\rangle$ and $|1\rangle$, respectively. It is not possible to exactly find the quantum state of qubit, i.e., α and β estimates. When qubit is gauged, the result comes as 0 having probability $|\boldsymbol{\alpha}|^2$ or 1 having probability $|\boldsymbol{\beta}|^2$. Therefore, $|\boldsymbol{\alpha}|^2+|\boldsymbol{\beta}|^2=1$ as sum of the probabilities has to be 1 [15]. However, we can use a scheme to measure qubit states as qubits can also be accepted as physical systems.

As an illustration, if $|0\rangle$ and $|1\rangle$ states can be associated with electron states of spin-down and spin-up, respectively, when present in a magnetic field, then the estimate of qubit state as per this foundation will be an estimate of the system's energy [17]. However, we must also keep in mind that the measurement action will demolish the superposition states (i.e., amplitudes), and we have either 0 or 1.

In the case of multiple qubit systems, the principles of quantum entanglement will also be applicable. It means when a set of qubits are matched up with each other, then action on any one influences the combined state of all. For a two-qubit system, the possible states would be the superposition of any of the following states denoted by $|00\rangle$, $|01\rangle$, $|10\rangle$, and $|11\rangle$. Superposition can be represented by the bell state equation as follows [15]:

$$|\psi\rangle = \frac{1}{\sqrt{2}}(|10\rangle + |01\rangle) \tag{2}$$

In the two-qubit system, if anyone measures the first qubit, it can only detect any of the states $|0\rangle$ and $|1\rangle$, with a chance of 50 percent. It is similar to the measurement of a single-qubit scenario.

Now, for the explanation, let us assume that the result of the first qubit's measurement is $|0\rangle$. Then the system settles to $|01\rangle$. Thus estimation of the second qubit gives $|1\rangle$, a result with a chance of 100 percent. Similarly, if the result of the first qubit is $|1\rangle$, the estimation of the second qubit will give the result $|0\rangle$.

We can see that any action (here estimation with result '0') executed on the first qubit influences the results of the assessment of the second qubit. When we generalize with n-qubit system, the computing rudimentary states for such a system will have binary string form $|x_1x_2..x_n\rangle$, where $x_{i(i=1,2,..n)}$ can be 0 or 1. Therefore, for this n-qubit system, the possible states would be superposition of any of the following 2^n states denoted by $|0\ldots 0\rangle$ till $|1\ldots 1\rangle$.

Quantum Information and Their Properties

Quantum computers have several advantages using the amalgamation of three significant characteristics — quantum interference, superposition, and entanglement. The combination of these three characteristics is substantial in a particular case and is observed to be unique to quantum mechanical systems.

Quantum Interference

This interference provides the outcome due to the amplitudes among two quantum states which are subtracted or summed. It might result in destructive and constructive interference [1, 18]. Some examples can be shown in the single-qubit quantum gate and its repeated application, known as the hadamard gate [19].

Quantum Entanglement

Quantum Entanglement is a unique computational resource in the view of computation. It occurs when a cluster of particles are generated and they interact. Finally, these particles share spatial closeness so that the quantum state of every group particle cannot be explained separately from the state. It is also unique to quantum computational systems. Researchers have demonstrated that it has multiple theoretical quantum advantages [1, 20].

Perhaps the one of the known examples of entanglement is the set of four 2 qubit bell states, $\{|\Phi^+\rangle, |\Phi^-\rangle, |\Psi^+\rangle, |\Psi^-\rangle\}$. It corresponds to the four maximally entangled states for two qubits [1]. For example, the given bell state

$$|\Phi^+\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle) \tag{3}$$

A dimension of the first-qubit state instantly implies knowledge of the second-qubit state.

The 'quantum entanglement phenomenon' is the cornerstone of quantum calculation speed and pieces of literature are available [21, 22]. It has been demonstrated that if the 'quantum entanglement phenomenon' is not used in the quantum algorithm, then the same algorithm can be worked with classical 0 and 1 bits computer systems without any remarkable change in computational speed. As an illustration, in the n-qubit system, the 'quantum entanglement phenomenon' empowers each of the amplitudes to be free, which yields a 2^n spatial vector system. Hence, it can be seen that quantum computation algorithms can perform the calculation at much higher-order speeds than their classical counterparts.

Superposition in Quantum Computing

A quantum particle can be present in two separate states together. However, the two quantum states or more can be "superposed" or added simultaneously. In this direction, multiple qubits can be superposed as a linear amalgamation over a foundation set, known as a coherent superposition. It is one of the fundamental properties of quantum mechanics. One of the examples is in the case of two qubits on the ideal basis [1], which is as follows:

$$|\psi\rangle = \alpha|00\rangle + \beta|01\rangle + \gamma|10\rangle + \delta|11\rangle \tag{4}$$

Quantum Gates: Quantum Logic Gates for Quantum Computing

Quantum gates are a pivotal piece of the quantum computing technology and play an essential role in the progress of quantum computers. They are a form of operation implemented on qubits, the crucial components of quantum computing. A quantum gate is a unitary transformation that works on either one or multiple qubits and changes the qubits' state. Quantum gates are utilized to govern the evolution of a quantum system and are thus used to execute various operations on a quantum computer. They are used as the building blocks of quantum algorithms and to manipulate qubits and perform operations on them. Combining quantum gates in different ways makes it possible to create a wide selection of quantum algorithms [23]. These gates can be understood with abstract operations [15], yet quantum mechanics postulates that they make it necessary for these operations to be shown in the form of the unitary matrix. These linear transformations maintain the coordination of the system. To be more specific, applying a quantum gate in an array having 'n' number of qubits is the same as performing multiplication to a $2^n \times 2^n$ matrix and 2^n entry vector. This capacity of the quantum computers is to gather, store, and operate on calculations having large amounts of data, i.e., 2^n, while manipulating only a few elements, say 'n' that makes it so powerful as compared to the classical computers [23].

In basic terms, a quantum gate is an action that can be taken on a qubit system. The laws of quantum mechanics set two precise prerequisites for how quantum gates must be formed. Firstly, linear quantum operators must be considered. This linear phenomenon is an important mathematical concept that has a significant impact on the nature of the physics of quantum systems and operations and their usefulness in computing. If, as an illustration, we consider one linear operator $\hat{O}$ and apply it to a mix of states, the outcome is the combination of all the affected states after being acted upon by this operator. Concerning one single qubit, it translates into as follows [15]:

$$\hat{O}(\alpha|0\rangle + \beta|1\rangle) = \alpha\left(\hat{O}|0\rangle\right) + \beta\left(\hat{O}|1\rangle\right) = \alpha'|0\rangle + \beta'|1\rangle \quad (5)$$

where α' and β' are complex numbers associated with the quantum state after the qubit gate has functioned. Operators of the linear form can be shown as matrices, i.e., tables demonstrating linear execution on every base state. However, all matrices does not represent valid quantum gates. It is anticipated that a quantum gate should put on an array of qubits, giving rise to another legitimate accumulation of qubits, particularly one being normalized. This condition is satisfied when the matrix symbolizing a quantum gate must be unitary i.e., $U^*U = UU^* = I$. Here, U^* represents the same matrix U in which columns and rows are interchanged, and each complex number is the conjugate of the previous one (in other words, each imaginary part has a negative factor). I is a 2×2 identity matrix. Any $2^n \times 2^n$ unitary matrix represents a legitimate quantum gate associated with n qubits [15].

In conventional computing, the only single-bit gate available is the NOT gate, which transforms 0 to 1 and vice versa. However, in quantum computation, a vast amount of 2×2 unitary matrices can be used as a qubit quantum gate of a single form. Researchers tried to understand the X or quantum-NOT gate in quantum circuits from time to time (Fig. 3). The quantum circuit is well studied. In the quantum circuit, the bell state was generated ($\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$) using the hadamard gate (Fig. 4). One of the earliest achievements in quantum computation is the realization of the possible options that can be enacted using a unique set comprising the universal gates for one qubit or two qubits [24, 25]. To put it simply, given any quantum gate, it can be implemented with a circuit constructed with one- and two-qubit gates with high accuracy [23]. However, more accuracy is needed to perform efficiently. Approximation of most of the quantum gates can only be achieved with the enormous quantity of gates taken from the universal set [15], meaning that implementing them would take an exponential amount of time, thus negating any quantum advantage. Classical gates are logic gates that manipulate binary values using boolean logic. These are used in digital electronic circuits to perform logical operations, such as AND, XOR, and NAND [19]. Classical gates are the basis of most digital systems today. Quantum gates, on the other hand, are quantum versions of classical gates. They use quantum mechanical phenomena such as superposition and entanglement to manipulate qubits, which are quantum versions of the classical bits used in classical gates. Quantum gates are used in quantum computers to perform quantum operations and computations. The most fundamental contrast between the classical and quantum gates is the way that they manipulate information. Classical gates work with binary values using boolean logic. Quantum gates, however, manipulate qubits using quantum mechanical phenomena. It allows quantum gates to operate on much larger datasets than classical gates and to perform operations much more quickly [23].

Quantum gates can also perform operations that classical gates cannot, such as teleportation, entanglement, and superdense coding. These operations are essential for performing quantum computations, which can be used to solve complex problems that are intractable for classical computers. In short, quantum gates can perform operations that classical gates cannot and can process much larger datasets than classical gates [23, 26]. It makes them immensely powerful for solving complex problems than classical computers. Quantum gates can also be used to create secure communication networks. They can protect data from being intercepted and

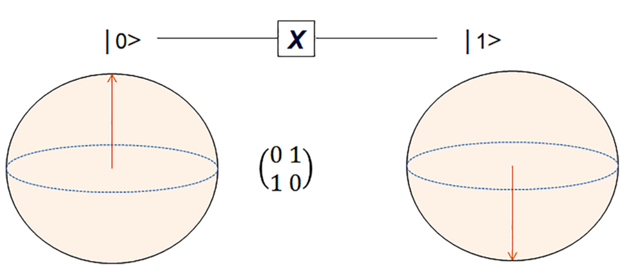

Fig. 3 The figure depicts a quantum circuit implementing the X or quantum-NOT gate

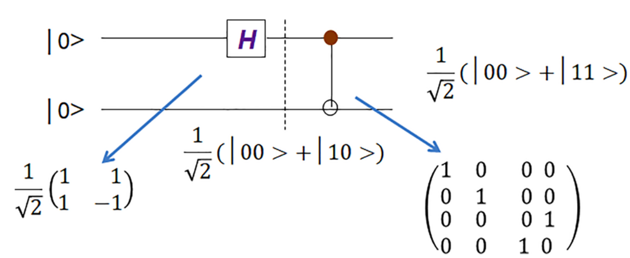

Fig. 4 In the quantum circuit, the bell state was generated ($1/\sqrt{2}$ ($|00\rangle$ +$|11\rangle$)) using the hadamard gate. It also shows the controlled-NOT gate

decrypted by employing quantum cryptography, which relies on the principles of quantum mechanics. Quantum gates are essential for the development of quantum computing, and their use will become more widespread as the technology advances. The most common type of quantum gate is the CNOT gate. This gate performs two-qubit operations, in which one qubit acts as control and the other qubit acts as a target. Other standard quantum gates include the hadamard gate, the toffoli gate, and the phase gate [15, 19, 26].

Quantum Hardware for Quantum Computer

Quantum hardware is an emerging field in advancing technology that can transform how we think about computing. For quantum algorithms to tackle challenging problems, they must be executed on suitable quantum hardware. Quantum hardware describes the physical systems and components of the quantum computer [23]. These components are usually made up of superconducting material, which allow them to operate at extremely low temperatures. The essential quantum hardware components are the qubits, among the simplest units of knowledge and information in the quantum computer. To create a qubit, physicists must first cool down a material to near absolute zero [27]. Then, they must carefully apply a magnetic field to trap a single electron in a tiny orbit around the atom's nucleus. This process is called quantum confinement. Once the qubit has been created, it can be manipulated using various techniques. For example, the strength of the magnetic field can be increased or decreased to change the qubit's spin. By carefully controlling the qubit's spin, a physicist can store a single quantum bit of information [28]. Conventional computers utilize binary bits to represent knowledge and information. However, a quantum computer uses qubits, which are the quantum bits that can exist in multiple states simultaneously. It allows for manipulating information at the quantum level, leading to unprecedented speed and accuracy. In contrast, a classical computer would require multiple bits to store the same data [23].

An unavoidable issue in all of these approaches is the presence of errors while computing, which could drastically impair the quality of the results. Thankfully, quantum error correcting codes have been discovered to reduce the impact of these errors, although they come with the drawback of needing an excessive number of qubits. Consequently, many engineering advances are still required for the system to tolerate faults [23]. A multitude of factors can lead to mistakes on a quantum computer, such as decoherence. Even subtle alterations can cause the intended quantum gate to be shifted, and the results will be dissimilar to what was expected. Additionally, the imperfect control systems of these machines will still cause a certain number of errors. Currently, the least prone error to quantum gates has been found in trapped-ion processors, with single-qubit gate errors of one part at 10^6 and two-qubit gate errors of 0.1% [29, 30]. As an illustration, a type of superconducting processor used for Google's quantum supremacy experiment had single-qubit and two-qubit gate errors of 0.1% and 0.3%, respectively [5].

A single malfunction in a gate could ruin a delicate calculation and thus, errors can cause the computation to become useless after a handful of gate sequences. As part of the quantum computing process, constructing codes to correct errors in quantum computation has been a critical factor. In the late nineties, several organizations and groups demonstrated codes that could produce fault-free computing as long as the gate errors stay under a specific boundary that depends entirely on the particular code [31]. The surface code, one of the most frequently used techniques, can operate with error rates close to 1% [32]. Sadly, codes to correct the quantum errors necessitate plenty of actual tangible qubits to encrypt a theoretically analytical form of the qubit, which is used for the computation process. As an illustration, a quantum algorithm used for factorizing prime numbers can, in a setting free of any noise, factorize a number of 2,000 bits with qubits ranging to almost 4,000. This factorization can be undertaken with an assumption of a 16-GHz gate rate, which takes approximately one day to complete [33]. Presently available technology of 433 qubit quantum processor [34] can perform computations that still need to be more to realize the full benefit of quantum computing. Hence, there is ample scope to develop the technology further.

Several teams have sought to establish unique algorithms for producing quantum processors at an intermediate level [35] that are subjected to noise. For instance, the variational algorithms employ a classic computer together with a smaller quantum processor to complete many short forms of quantum computations before noise disrupts them. These established algorithms incorporate the parameterized circuits of quantum that carries out a challenging task, and the classical computer optimizes these frameworks. Error mitigation is an essential technique that attempts to reduce the number of errors with a minimal effort to operate larger circuits instead of reaching fault tolerance. Methods for doing this include discarding runs with errors through additional operations or handling the error percentage to arrive at the correct conclusion [36]. Even though the most significant applications will need unimaginably huge and resilient quantum computers, it is predicted that the quantum devices available in the coming years will ultimately be capable of accomplishing practical tasks [35].

In addition to quantum computers, there are other forms of quantum hardware. These include quantum sensors which can detect and measure physical phenomena at the quantum level and quantum communication systems which can securely transmit information over long distances [37]. The developmental process to build successful quantum hardware is in the early stages, but until now, great promise has already been shown in this field. In future, quantum hardware could be used to solve complex problems beyond traditional computers' capabilities. It could also be used to create new forms of communication and encryption and to develop new areas in artificial intelligence. In conclusion, quantum hardware is a fast-developing technology with enormous potential to change how we think about computing. As technology develops, quantum hardware will likely become integral to our lives.

State-of-the-Art Quantum Processors

Several quantum processors are available in the superconducting architecture that IBM predominantly manufactures. Xanadu Quantum Technologies also manufactures some quantum processors. Intel quantum processors are also available. The processing units in quantum computers are called quantum processing units (QPUs). In the quantum computing system, different ranges of qubit quantum processors are available (Fig. 5), and the qubit (Q) of the processor ranges are noted from 2 to 433 qubits. 2 qubit processor is developed by the Delft University of Technology (known as TU Delft). At the same time, 433 qubit processor is designed by IBM in the IBM Osprey architecture. Recently, a startup company from China known as SpinQ Technology informed about the development of the first-desktop quantum computer with a high-speed processor [8, 19, 38, 39].

Quantum RAM (qRAM) and Big Data Analysis

Presently, quantum computer models are unable to access large classical datasets during the superposition. Therefore, current quantum computers have less advantage in unraveling problems using large datasets. However, a tentative possibility for a quantum hardware solution is qRAM. It might provide the ability to coherently enquire a big dataset as a superposition of qubits in a quantum computer. The output yield would be composed of a superposition of the contents of the memory cell [8].

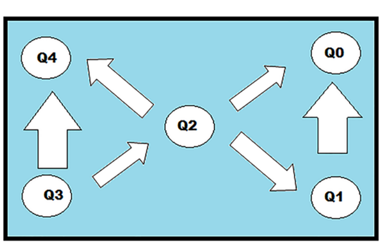

Fig. 5 The figure depicts a 5 qubit quantum processor layout in a chip

Quantum Annealing (QA) Devices

The device (QA device) provides an opportunity for another approach to quantum computing with noisy intermediate-scale quantum (NISQ) hardware (Fig. 6). NISQ hardware is specialized based on classical simulated annealing. It might provide several quantum computing advantages, such as simulation and optimization [1]. QA has been explained by researchers using available QA hardware. A single-nucleotide sequence model's feature has been solved using QA devices [40, 41].

Quantum Algorithms

We are aware of the capability of quantum computers which can be used in solving various jobs [32, 42–45]. To understand 'how fast' these interesting machines are [46], one needs to comprehend the differences between the classical algorithms with the quantum ones. However, it is to be noted that in many situations, researchers cannot decide the best achievable quantum algorithm.

Several algorithms have tried to implement in the quantum computation platform, like the classical computations, such as ensemble methods, persistent homology, k-Means clustering, boltzmann machines, and gaussian process regression. For quantum computation, the algorithms have used the quantum computation in σ(N) and related factors. It has been noted that some algorithms use quantum random access memory (QRAM), such as k-Means clustering, gaussian process regression, and gaussian mixture models (Table 1) [23].

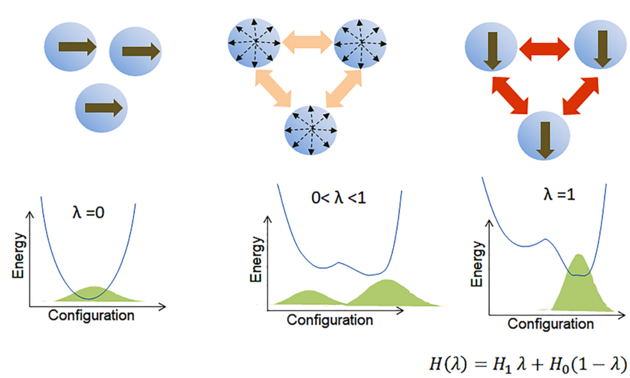

Fig. 6 The figure explains the quantum annealing process. In this process, the qubits are physically evolved from a situation run by a Hamiltonian H0. For H0, the ground state is known a situation governed by H1

Table 1 Synopsis of some essential quantum computing algorithms with their features

Sl. No	Algorithm type	Classical computation in *o(N)* and related factors	Quantum computation in *o(N)* and related factors	Quantum random access memory (QRAM)	References
1	Ensemble methods	$o(N)$	$o(\sqrt{N})$	No	[87–89]
2	Persistent homology	$o(expN)$	$o(N^5)$	No	[90]
3	Hidden Markov models	$o(N)$	Unclear	No	[91, 92]
4	k-Means clustering	$o(kN)$	$o(log\ kN)$	Yes	[93–95]
5	Bayesian deep learning	$o(N)$	$o(\sqrt{N})$	No	[96]
6	Boltzmann machines	$o(N)$	$o(\sqrt{N})$	No	[97–101]
7	Gaussian process regression	$o(N^3)$	$o(log\ N)$	Yes	[102, 103]
8	Variational autoencoder	$o(expN)$	Unclear	No	[104]
9	Support vector machines	$\approx o(N^2)$-$o(N^3)$	$o(log\ N)$	Yes	[105–107]
10	Linear regression	$o(N)$	$o(log\ N)$	Yes	[108–111]
11	Graphical models	$o(N)$	Unclear	No	[112]
12	Principal component analysis	$o(N)$	$o(log\ N)$	No	[113]
13	Multilayer perceptrons	$o(N)$	Unclear	No	[114–116]
14	Bayesian networks	$o(N)$	$o(\sqrt{N})$	No	[117, 118]
15	Gaussian mixture models	$o(log\ N)$	$o(polylogN)$	Yes	[119, 120]
16	Decision trees	$o(N\ log\ N)$	Unclear	No	[120]
17	Generative adversarial networks	$o(N)$	$o(polylogN)$	No	[121, 122]
18	Reinforcement learning	$o(N)$	$o(\sqrt{N})$	No	[123]
19	Convolutional neural networks	$o(N)$	$o(log\ N)$	No	[124]

These algorithms have been reported from time to time in the literature

An exception to it is the best achievable quadratic speedup on search problem, which has already been demonstrated with grover's algorithm [41]. However, the knowledge from scenarios elucidates that 'quantum speedup' has not been validated.

Search Algorithm in the Quantum Computation

A vital advantage of the quantum computer over its classical counterpart is its extraordinary database searching speed [19]. An exciting algorithm on 'search' proposed by Grover illustrates this superiority. This algorithm can quadratically quicken a search problem of unstructured nature. It can play the role of 'subroutine or trick' to acquire enhancement in run time (quadratic form) in several different algorithms. This concept is known as the 'amplitude amplification trick.' When there is a need to determine marked items in a lengthy listing, 'quadratic speedup theory' as present in the grover algorithm, plays a crucial role in order to save time. In addition, this algorithm does not utilize the lengthy list's inner form; hence, it can be categorized as generic.

Grover's algorithm is usually considered as a helpful one for database searching. Sometimes, grover's search shows worse performance than shor's algorithm. However, srover's algorithm is used to solve problems with appropriate oracles. The algorithm is applied to find maximal cliques, finding cycles, and finding triangles in a graph [47],

Consequently, for various classical problems, the instantaneous quadratic speedup can be obtained with its use. Grover's algorithm determines one specific output out of N options in the case of queries [41]. Interestingly, this particular output is optimal [48]. It can act as a subroutine of many complex algorithms used in computational biology for data processing, such as advanced proposition in protein sequence comparison [49], the neuronal version to understand the intracellular Ca^{2+} dynamics [50], and leading algorithms in the quantum domain of machine learning [51].

Discrete and Factorization Algorithm

It can be recalled that an established and effective quantum algorithm is available for problems dealing with integer factorization and discrete logarithms [47], providing remarkable computational speed over the most significant classical algorithm. Even though an integer can be decomposed as a 'product of primes,' determining the prime factors can be challenging. As an illustration [47], it may be noted that the safety of many internet transactions depends on the following theory: factoring integers possessing a huge number of digits (say, ≥ thousand) is almost impossible. Shor [52] challenged this theory and demonstrated a quantum algorithm with polynomial time. Researchers developed the algorithm to determine the prime factors of an integer N and executed it in polynomial time. Further, in the presently employed cryptography algorithms, this algorithm [52] has potential use in performing cryptanalysis where public-key cryptography could be broken. With this background, we note that quantum technology has not yet developed to that extent of solving and implementing these types of problems in totality. Nevertheless, future researchers can take benefit of Shor's algorithm in which 'subroutine parts' can be considered. One such example of the 'subroutine part' is the 'Quantum Fourier Transform,' which has potential use in various quantum algorithms as 'subroutines' [41].

The Reliability and Validity Studies Proved the Efficiency and Effectiveness of Quantum Computing and its Qubit

Only some studies conclusively demonstrated the effectiveness and efficiency of quantum processing and qubits, despite ongoing research and development in the field. Tannu and Qureshi tried to understand the variability of the two qubits. These researchers found a Noisy system with a few hundred qubits, entitled NISQ (Noisy Intermediate-Scale Quantum computers). They observed variations in the error rates of different qubits [53]. Cross et al. tried to validate the quantum computers using randomized model circuits and have attempted to introduce a method for NISQ [54]. Similarly, Piveteau et al. mitigated the error for universal gates on programmed qubits [55]. The necessity for accurate and effective error correction, the game-changing potential for energy applications, and the developing ecosystem and industry around quantum computing are all topics covered in papers and articles that explore the drawbacks and possible advantages of quantum computing. All national governments have made significant investments in experimental research to create scalable qubits with longer coherence durations and reduced error rates because it has proven challenging to construct high-quality qubits physically.

Applications of Quantum Computing to Understand the Next-Generation Biological Problems

Presently, considerable interest is increasing in quantum computing to solve biological problems [56–61]. Researchers are currently trying to solve next-generation biological problems using the quantum computing approach. Quantum computing devices are used to solve biologically significant problems. Quantum computing devices include NISQ and fault-tolerant quantum computing (FTQC) devices which are very much usual to solve biological problems (Table 2). Some biological problems which are being solved by the quantum computing approach are described as follows:

Table 2 Applications of quantum computing in computational biology and the probable impact from noisy intermediate-scale quantum (NISQ) and fault-tolerant quantum computing (FTQC) devices for usual next-generation biologically significant problems

Sl. no	Devices type	Applications in different aspects of molecular biology				
		Protein folding	Genome assembly	DNA-binding transcription factor	Quantum biophysics simulation	Quantum biochemistry simulation
1	Fault-tolerant quantum computing (FTQC) devices	Compared against the classical methods by applying quantum annealers of new generations programmable quantum simulators and gate-based advanced processors used for speedup	Accelerate the process in respect of classical methods. It used programmable quantum simulators and new generations of quantum annealers, along with the gate-based processors	Faster than classical methods by support of progressive gate-based processors, quantum annealers of generations, and programmable quantum simulators	In behind of classical simulations the simulation of biological problems is accomplished	Resolving the appropriate quantum chemistry problems which are beyond of classical simulations
2	Noisy intermediate-scale (NISQ) quantum devices	Small-scale problems are solved using the quantum annealers and demonstrations of the proof of concept	NISQ devices used to resolving small-scale problems by the quantum annealers and showing of proof of concept	Showing minor advantages compared with classical methods using quantum machine learning for the simplified datasets	For biophysical systems, the prototyping-oriented digital quantum Simulation algorithms have been performed	Applied to understandings about the configuration of simple molecules, for example, using of variational quantum algorithms (VQAs)

Simulation and Modeling of Biological Macromolecules

Researchers are using the present classical approaches to understand the simulation and modeling of biological macromolecules. Simulation is one of the significant interfaces between biology, chemistry, and physics. The simulation might provide a deeper insight into biological macromolecules, such as the association of small molecules with macromolecules, computation of ligand's binding free energies, dynamics of transport and ion channels across membranes, and the information might be the basis of clinical features. The study might contribute to the structure and function relationship of biological macromolecules [62]. Quantum computers have shown their promise in fastest problem solving and simulation [23].

The quantum simulator's primary role is to divulge details of a conceptual mathematical function associated with a physical model [51]. However, it must consider the particular objective and setting of that simulation. Generally, a simulation indicates if the model correctly represents the system under study. When the simulation model's depiction is precise, the given quantum simulator can be accepted for our system. The simulator definition can be related to established scenarios where the term has been applied. Several devices that are publicized in the name 'quantum simulation' are, in fact, analog simulators [63–67]. Hamiltonians of these devices can be manipulated and used in expressing real systems. This tallies our explanation of the simulator and the context in which it is discussed above. Thus quantum simulators can be regarded as a 'universal quantum computer.' It is noteworthy that we can program quantum computers to simulate localized quantum systems [68]. This capability indicates that it can encode quantum mechanical entities in qubits and gates. Simultaneously, the current scientific belief is that quantum systems cannot be accurately simulated with classical computers. Therefore, quantum computers can play a crucial role in performing precise simulations for chemical processes and procedures in biological sciences [9, 69–71].

Computational Biology and Quantum Computing

Bioinformatics performs a key role in understanding and optimization of the different computational tasks, like de novo assembly, phylogenetic tree inference, and sequence alignment. For these problems, classical algorithms are often used. However, these areas can be studied with quantum algorithms. Recently, quantum algorithms can be used to solve some bioinformatics problems (Table 3). Researchers target NP-hard problems in FTQC devices for theoretical algorithms of quantum computing. One example includes sequence alignment [49, 72]. Prousalis and Konofaos performed a quantum pattern recognition method to develop advanced pairwise sequence alignment. However, in terms of time and space complexity, the proposed method displayed a better alignment quality and succeeded among the others [49]. Another example is phylogenetic tree development using a quantum-inspired computer. Phylogenetic tree development methods are a crucial area of evolutionary biology, which is fundamental for many biological science studies. It presents ample information on evolutionary events among organisms [73]. Recently, Onodera et al. reconstructed a phylogenetic tree using graph cut. In this study, the researchers used a quantum-inspired computer. However, current research in quantum computing shows significant promise in bioinformatics [74].

Data Analysis in Bioinformatics

It is a popular fact that linear and differential equations have broad applications in numerous engineering and other scientific domains. Recent research trends indicate 'data sets' which are used to define these equations, which can be enormous depending on the specific application [75], where data of the order petabytes could be required to process to get a solution. Quantum computation has the power to process magnanimous data sets at exponential speeds and can feed data to AI techniques. Several researchers have used quantum computation in data analysis. Available pieces of literature reveal the application of quantum algorithms to solve linear [75] and differential equations [76–78], where these algorithms play the role of 'subroutines' in algorithms on data processing. As proposed by Harrow et al. [75], the quantum algorithm is capable enough to give solutions to definite linear systems at exponentially rapid speed compared to any existing algorithm. This group [75] reported that their algorithm was related to the traditional Monte Carlo ones, which are pretty quick and operate with samples of a statistical distribution involving N objects instead of using all the N constituents of distribution. They proved that even though these traditional sampling ones are quick, any traditional algorithm, in general, needs much more time than their [75] algorithm to complete the identical matrix inversion job. On the specific applications of this algorithm [75], its use is possible in computational biology, where numeric models can be solved, and also in the domain of machine learning techniques.

Protein Folding

The protein-folding problem is quite old and was proposed one half-century ago. The protein-folding problem is one of the significant and complex problems in computational biology [79]. Researchers tried to predict the protein's three-dimensional structure using a given amino acid sequence. However, the protein-folding problem has been studied

Table 3 Some significant quantum computation approaches to address complex biological problems. It illustrates the next-generation computational biology landscape

Sl. No	Types of biological application	Type of algorithm	Hardware	Experimental validation	Classical complexity	Remarks	References
1	Sequence alignment	Optimization	Universal gate-based quantum device	Not available	Experimental estimate and polynomial	Expected advantage in polynomial	[125–129]
2	Inference of phylogenetic trees	Optimization	Universal gate-based quantum device	Not available	Super- polynomial	Predictable advantage in polynomial	[130]
3	Molecular docking simulation	Sampling	Gaussian boson sampler	Not available	Super- polynomial	Up to super-polynomial in unknown state	[131]
4	Neural networks	QML	Universal gate-based quantum device	Completed	Super-polynomial and polynomial (Boltzmann machine)	Predictable advantage in polynomial and problem specific, which differs by measure	[132–136]
5	Inference of biological networks	Optimization	Quantum annealer	Completed	Super-polynomial and polynomial	Predictable advantage in polynomial	[137–139]
6	Conformation simulation and folding of protein	Quantum annealing	Quantum annealer	Completed	Experimental estimate and polynomial	Up to polynomial in unknown state	[23, 80, 140–142]
7	Sequence matching	Search, QML	Universal gate-based quantum device	Not available	Polynomial	Advantage probably up to super-polynomial	[136, 143, 144]
8	Transcription factor-binding analysis	Optimization	Quantum annealer	Completed	Experimental estimate and polynomial	Expected advantage up to polynomial in unknown state	[145]
9	De novo assembly	Quantum annealing, optimization	Universal gate-based quantum device, quantum annealer	Completed	Experimental estimate and polynomial	Probably up to polynomial in unknown state	[146, 147]

intensively in quantum computers. Researchers reformulated the lattice problem to make the issue of protein folding compatible with the present day's quantum computers. Several lattice protein-folding models were developed from time to time. In this direction, using Hamiltonian was studied. The model is a specific example of a thermodynamic system that can understand phase transitions. The problem was attempted to slove using quantum annealing [80]. Perdomo–Ortiz et al. used quantum annealing to find low-energy conformations of different lattice protein models [81]. The study was standard execution of quantum annealing to solve the lattice protein-folding problems. In this direction, several algorithms have been developed from time to time. One significant algorithm is Quantum Approximate Optimization Algorithm (QAOA). However, machine learning models in quantum computing might speed up solving the protein-folding problem.

Molecular Biology Problems

One of the significant areas in molecular biology is transcription factor-binding sites in DNA. These factors are essential in regulating gene expression. Researchers tried to identify the different transcription factors for DNA binding using quantum computing. Li et al. used quantum annealing to predict binding specificity using a small number of DNA sequences datasets derived from actual binding affinity experiments [82]. The researchers described that quantum annealing is as an efficient technique to solve computational biology problems in quantum computing.

Modeling of Gene Regulatory Networks

Modeling of gene regulatory networks is a fascinating area in quantum computing. To perform modeling of gene regulatory networks, Weidner et al. have created a quantum circuit from a series of quantum gates using the development network of the mammalian cortical area. Here, the circuit executes a state transition on the superposition of the identical and finally, they have measured the output [83].

Drug Discovery and Development

Quantum computing is exploring in the field of drug discovery and development. At the same time, quantum computing can be used in the manufacturing, supply chain, and other use of drug distribution. Zinner et al. proposed a plan for using quantum computing along with the CADD and AI during drug discovery, development, and distribution [4]. Recently, Lau et al. enriched the drug discovery and development area by inventing a hybrid classical quantum workflow model for drug design entitled HypaCADD. This model incorporates quantum machine learning (QML) with classical docking and molecular dynamics. Finally, they presented a case study with SARS-CoV-2 protease and related mutants using the model [84].

Mechano-Biology

Quantum computing is also being applied in mechano-biology. Recently, Sohail and Ashiq have tried to use a model for quantum-inspired AI for the sensor development of a cardiac problem, like cardiac amyloidosis. It is a good example of quantum computing in the field of mechano-biology [85].

RNA Folding

Understanding an RNA-fold pattern is an exciting area of biology. Recently, Fox and his colleagues have shown that the RNA secondary structure prediction issue can be solved as Binary Quadratic Model, which can be addressed using quantum computing [86]. Soon, more applications are yet to come, which might solve the RNA secondary structure prediction issue more profoundly.

Challenges and Limitations of Quantum Computing

Although quantum computing is a new technology with immense promise to address complicated problems, it also has several difficulties and constraints. Due to their extreme sensitivity to environmental disturbances and the requirement for expensive refrigerators with temperatures close to absolute zero, quantum computers provide one of the key hurdles. Because of this, they are unstable and challenging to use. Error correction is a severe problem for quantum computers because they are noise sensitive and difficult to calibrate. Because qubits can exist in an endless number of states, unlike conventional computers that undergo a bit flip from 0 to 1 or vice versa, quantum faults are more challenging to fix. The fact that quantum computers are still being developed and still need to be economically viable presents another challenge. With only 70 qubits, current machines fall well short of the one-million qubits required to make quantum computers economically viable. Such a breakthrough, according to researchers, may occur within the next ten years, but in the meantime, quantum computers will continue to be costly and specialized devices. The hardware and software limitations of quantum computing also have their limits. Quantum computing relies on quantum gates, which alter data and apply logical operations to electronic signals.

A quantum logic gate or quantum gate is an elemental quantum circuit operating on a small number of qubits. It is formal and foundational for quantum mechanics based on modifying some rules. Quantum gates cannot store data, though, which makes it challenging to scale quantum computing to more complex issues. Furthermore, strictly quantum algorithms are challenging to implement, especially with current hardware.

Conclusion

The field of quantum computing is passing through main phases of development. The software and hardware in the field of quantum computing are progressing. However, there are considerable knowledge gaps in this area which remain as the main challenges. However, more than hundred organizations, including research, academic, government laboratories, and computer organizations worldwide, are working to address these challenges to fill the knowledge gaps. Presently high-end ion-trap quantum computers are expected to develop very soon with more than 450 to 500 qubit processor with highly dynamic features. It will handle the larger data set sizes more prominently. The new quantum algorithm development for next-generation biological problems is one of the significant challenges in quantum computing in computational biology. However, computational biologists will probably have substantial challenges. Therefore, our review will enrich the computational biology field and help the future researchers to solve the significant, modern molecular biological problems for the future generation very easily.

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/s12033-023-00765-4.

Author contributions SP: contributed to validation, formal analysis, investigation, and writing of the original draft. MB: contributed to figure development and data curation, resources. S-SL: contributed to investigation and validation. CC: contributed to conceptualization, investigation, methodology, project administration, and writing, reviewing, & editing of the manuscript.

Data availability All data included within the manuscript.

Declarations

Conflict of interest The authors declare that they have no competing interests.

Ethical Approval Not applicable.

Consent to Participate Not applicable.

Consent for Publication Not applicable.

References

1. Cordier, B. A., Sawaya, N. P. D., Guerreschi, G. G., & McWeeney, S. K. (2022). Biology and medicine in the landscape of quantum advantages. *Journal of the Royal Society Interface, 19*(196), 20220541.
2. Graham, F. (2021). Daily briefing: Biology begins to tangle with quantum computing. *Nature*. https://doi.org/10.1038/d41586-021-01856-5
3. Marchetti, L., Nifosi, R., Martelli, P. L., Da Pozzo, E., Cappello, V., Banterle, F., Trincavelli, M. L., Martini, C., & D'Elia, M. (2022). Quantum computing algorithms: getting closer to critical problems in computational biology. *Briefings in Bioinformatics*. https://doi.org/10.1093/bib/bbac437
4. Zinner, M., Dahlhausen, F., Boehme, P., Ehlers, J., Bieske, L., & Fehring, L. (2022). Toward the institutionalization of quantum computing in pharmaceutical research. *Drug Discovery Today, 27*(2), 378–383.
5. Arute, F., Arya, K., Babbush, R., Bacon, D., Bardin, J. C., Barends, R., Biswas, R., Boixo, S., et al. (2019). Quantum supremacy using a programmable superconducting processor. *Nature, 574*(7779), 505–510.
6. Zhong, H. S., Wang, H., Deng, Y. H., Chen, M. C., Peng, L. C., Luo, Y. H., Qin, J., Wu, D., Ding, X., Hu, Y., Hu, P., Yang, X. Y., Zhang, W. J., Li, H., Li, Y., Jiang, X., Gan, L., Yang, G., You, L., … Pan, J. W. (2020). Quantum computational advantage using photons. *Science, 370*(6523), 1460–1463.
7. Cheng, H. P., Deumens, E., Freericks, J. K., Li, C., & Sanders, B. A. (2020). Application of quantum computing to biochemical systems: A look to the future. *Frontiers in Chemistry, 8*, 587143.
8. Emani, P. S., Warrell, J., Anticevic, A., Bekiranov, S., Gandal, M., McConnell, M. J., Sapiro, G., Aspuru-Guzik, A., Baker, J. T., Bastiani, M., Murray, J. D., Sotiropoulos, S. N., Taylor, J., Senthil, G., Lehner, T., Gerstein, M. B., & Harrow, A. W. (2021). Quantum computing at the frontiers of biological sciences. *Nature Methods, 18*(7), 701–709.
9. Cao, Y., Romero, J., Olson, J. P., Degroote, M., Johnson, P. D., Kieferova, M., Kivlichan, I. D., Menke, T., Peropadre, B., Sawaya, N. P. D., Sim, S., Veis, L., & Aspuru-Guzik, A. (2019). Quantum chemistry in the age of quantum computing. *Chemical Reviews, 119*(19), 10856–10915.
10. Kirsopp, J. J. M., Di Paola, C., Manrique, D. Z., Krompiec, M., Greene-Diniz, G., Guba, W., Meyder, A., Wolf, D., Strahm, M., & Muñoz Ramo, D. (2022). Quantum computational quantification of protein–ligand interactions. *International Journal of Quantum Chemistry, 122*(22), e26975.
11. Fox, D. M., Branson, K. M., & Walker, R. C. (2021). mRNA codon optimization with quantum computers. *PLoS ONE, 16*(10), e0259101.
12. Robert, A., Barkoutsos, P. K., Woerner, S., & Tavernelli, I. (2021). Resource-efficient quantum algorithm for protein folding. *npj Quantum Information, 7*(1), 38.
13. Dirac, P. A. M. (1939). A new notation for quantum mechanics. *Mathematical Proceedings of the Cambridge Philosophical Society, 3*, 416–418.
14. Horowitz, M., & Grumbling, E. (2019). *Quantum computing: progress and prospects*. The National Academies Press.
15. Nielsen, M. A., & Chuang, I. L. (2010). *Quantum Computation and Quantum Information: 10th Anniversary*. Cambridge University Press.
16. Schrödinger, E. (1935). Die gegenwärtige situation in der quantenmechanik. *Naturwissenschaften, 23*(50), 844–849.
17. Outeiral, C., Strahm, M., Shi, J., Morris, G., Benjamin, S., & Deane, C. (2021). The prospects of quantum computing in computational molecular biology. *Computational Molecular Science, 11*(1), e1481.
18. Liu, S.-X., Ismael, A. K., Al-Jobory, A., & Lambert, C. J. (2023). Signatures of room-temperature quantum interference in molecular junctions. *Accounts of Chemical Research, 56*(3), 322–331.
19. I.Q.C. Operations glossary, IBM Quantum. 2022., https://quantum-computing.ibm.com/composer/docs/iqx/terms-glossary.
20. Asif, N., Khalid, U., Khan, A., Duong, T. Q., & Shin, H. (2023). Entanglement detection with artificial neural networks. *Science and Reports, 13*(1), 1562.
21. D. Gottesman, The Heisenberg representation of quantum computers, arXiv preprint quant-ph/9807006 (1998).
22. Jozsa, R., & Linden, N. (2003). On the role of entanglement in quantum-computational speed-up. *Proceedings of the Royal Society of London Series A: Mathematical, Physical and Engineering Sciences, 459*(2036), 2011–2032.
23. 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.
24. Barenco, A., Bennett, C. H., Cleve, R., DiVincenzo, D. P., Margolus, N., Shor, P., Sleator, T., Smolin, J. A., & Weinfurter, H. (1995). Elementary gates for quantum computation. *physical Review A, 52*(5), 3457–3467.
25. Bremner, M. J., Dawson, C. M., Dodd, J. L., Gilchrist, A., Harrow, A. W., Mortimer, D., Nielsen, M. A., & Osborne, T. J. (2002). Practical scheme for quantum computation with any two-qubit entangling gate. *Physical Review Letters, 89*(24), 247902.
26. Sauvage, F., & Mintert, F. (2022). Optimal control of families of quantum gates. *Physical Review Letters, 129*(5), 050507.
27. D.L. Chandler, Ultracold molecules hold promise for quantum computing. MIT News https://news.mit.edu/2017/ultracold-molecules-hold-promise-quantum-computing-qubit-0727, (2017).
28. Sekiguchi, Y., Matsushita, K., Kawasaki, Y., & Kosaka, H. (2022). Optically addressable universal holonomic quantum gates on diamond spins. *Nature Photonics, 16*(9), 662–666.
29. Harty, T. P., Allcock, D. T., Ballance, C. J., Guidoni, L., Janacek, H. A., Linke, N. M., Stacey, D. N., & Lucas, D. M. (2014). High-fidelity preparation, gates, memory, and readout of a trapped-ion quantum bit. *Physical Review Letters, 113*(22), 220501.
30. Ballance, C. J., Harty, T. P., Linke, N. M., Sepiol, M. A., & Lucas, D. M. (2016). High-fidelity quantum logic gates using trapped-ion hyperfine qubits. *Physical Review Letters, 117*(6), 060504.
31. Gottesman, D. (1997). *Stabilizer codes and quantum error correction*. California Institute of Technology.
32. Fowler, A. G., Mariantoni, M., Martinis, J. M., & Cleland, A. N. (2012). Surface codes: Towards practical large-scale quantum computation. *Physical Review A, 86*(3), 032324.

33. P.W. Shor, Algorithms for quantum computation: discrete logarithms and factoring, In: *Proceedings 35th Annual Symposium on Foundations of Computer Science*, 1994, pp. 124–134.
34. I.N. IBM. (2022) IBM unveils 400 qubit-plus quantum processor and next-generation IBM quantum system two, IBM, November 9, 2022. https://newsroom.ibm.com/2022-11-09-IBM-Unveils-400-Qubit-Plus-Quantum-Processor-and-Next-Generation-IBM-Quantum-System-Two.
35. Preskill, J. (2018). Quantum computing in the NISQ era and beyond. *Quantum, 2*, 79.
36. Li, Y., & Benjamin, S. C. (2017). Efficient variational quantum simulator incorporating active error minimization. *Physical Review X, 7*(2), 021050.
37. Degen, C. L., Reinhard, F., & Cappellaro, P. (2017). Quantum sensing. *Reviews of Modern Physics, 89*(3), 035002.
38. Montenegro-Pohlhammer, N., Kuppusamy, S. K., Cardenas-Jiron, G., Calzado, C. J., & Ruben, M. (2023). Computational demonstration of isomer- and spin-state-dependent charge transport in molecular junctions composed of charge-neutral iron(II) spin-crossover complexes. *Dalton Transactions, 52*(5), 1229–1240.
39. Blunt, N. S., Camps, J., Crawford, O., Izsak, R., Leontica, S., Mirani, A., Moylett, A. E., Scivier, S. A., Sunderhauf, C., Schopf, P., Taylor, J. M., & Holzmann, N. (2022). Perspective on the current state-of-the-art of quantum computing for drug discovery applications. *Journal of Chemical Theory and Computation, 18*(12), 7001–7023.
40. Kokail, C., Maier, C., van Bijnen, R., Brydges, T., Joshi, M. K., Jurcevic, P., Muschik, C. A., Silvi, P., Blatt, R., Roos, C. F., & Zoller, P. (2019). Self-verifying variational quantum simulation of lattice models. *Nature, 569*(7756), 355–360.
41. Fedorov, A., & Gelfand, M. (2021). Towards practical applications in quantum computational biology. *Nature Computational Science, 1*(2), 114–119.
42. Montanaro, A. (2016). Quantum algorithms: An overview. *npj Quantum Information, 2*(1), 1–8.
43. Childs, A. M., & Van Dam, W. (2010). Quantum algorithms for algebraic problems. *Reviews of Modern Physics, 82*(1), 1.
44. M. Santha, Quantum walk based search algorithms, Theory and Applications of Models of Computation: 5th International Conference, TAMC 2008, Xi'an, China, April 25-29, 2008. Proceedings 5, Springer, 2008, pp. 31-46.
45. Bacon, D., & van Dam, W. (2010). Recent progress in quantum algorithms. *Communications of the ACM, 53*(2), 84–93.
46. S.W. Shin, G. Smith, J.A. Smolin, U. Vazirani, How" quantum" is the D-Wave machine?, *arXiv preprint* arXiv:1401.7087 *(2014).*
47. A. Adedoyin, J. Ambrosiano, P. Anisimov, A. Bärtschi, W. Casper, G. Chennupati, C. Coffrin, H. Djidjev, D. Gunter, S. Karra, Quantum algorithm implementations for beginners, *arXiv preprint* arXiv:1804.03719 *(2018).*
48. Bennett, C. H., Bernstein, E., Brassard, G., & Vazirani, U. (1997). Strengths and weaknesses of quantum computing. *SIAM Journal on Computing, 26*(5), 1510–1523.
49. Hollenberg, L. C. (2000). Fast quantum search algorithms in protein sequence comparisons: quantum bioinformatics, Physical review. *E, Statistical Physics, Plasmas, Fluids, and Related Interdisciplinary Topics, 62*(5), 7532–7535.
50. Clark, K. B. (2014). Basis for a neuronal version of Grover's quantum algorithm. *Frontiers in Molecular Neuroscience, 7*, 29.
51. Johnson, T. H., Clark, S. R., & Jaksch, D. (2014). What is a quantum simulator? *EPJ Quantum Technology, 1*(1), 1–12.
52. Shor, P. W. (1999). Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM Review, 41*(2), 303–332.
53. S.S. Tannu M.K. Qureshi, Not all qubits are created equal: A case for variability-aware policies for NISQ-era quantum computers. In: *Proceedings of the Twenty-Fourth International Conference on Architectural Support for Programming Languages and Operating Systems* (2019) 987–999.
54. Cross, A. W., Bishop, L. S., Sheldon, S., Nation, P. D., & Gambetta, J. M. (2019). Validating quantum computers using randomized model circuits. *Physical Review A, 100*(3), 032328.
55. Piveteau, C., Sutter, D., Bravyi, S., Gambetta, J. M., & Temme, K. (2021). Error mitigation for universal gates on encoded qubits. *Physical review letters., 127*(20), 200505.
56. Montenegro-Pohlhammer, N., Kuppusamy, S. K., Cardenas-Jiron, G., Calzado, C. J., & Ruben, M. (2023). Computational demonstration of Isomer-and spin-state-dependent charge transport in molecular junctions composed of charge-neutral iron (II) spin-crossover complexes. *Dalton Transactions., 52*(5), 1229–1240.
57. Khatami, M. H., Mendes, U. C., Wiebe, N., & Kim, P. M. (2023). Gate-based quantum computing for protein design. *PLOS Computational Biology., 19*(4), e1011033.
58. 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.
59. Swan, M., dos Santos, R. P., & Witte, F. (2022). Quantum Neurobiology. *Quantum Reports, 4*(1), 107–126.
60. Elaraby, A. (2022). Quantum medical images processing foundations and applications. *IET Quantum Communication, 3*(4), 201–213.
61. D'Acunto, M. (2022). Quantum biology π–π entanglement signatures in protein-DNA interactions. *Physical Biology, 9*(3), 036003.
62. van der Kamp, M. W., Shaw, K. E., Woods, C. J., & Mulholland, A. J. (2008). Biomolecular simulation and modelling: status, progress and prospects. *Journal of the Royal Society Interface, 5*(3), S173–S190.
63. Cirac, J. I., & Zoller, P. (2012). Goals and opportunities in quantum simulation. *Nature Physics, 8*(4), 264–266.
64. Hauke, P., Cucchietti, F. M., Tagliacozzo, L., Deutsch, I., & Lewenstein, M. (2012). Can one trust quantum simulators? *Reports on Progress in Physics, 75*(8), 082401.
65. Schaetz, T., Monroe, C. R., & Esslinger, T. (2013). Focus on quantum simulation. *New Journal of Physics, 15*(8), 085009.
66. Buluta, I., & Nori, F. (2009). Quantum simulators. *Science, 326*(5949), 108–111.
67. Georgescu, I. M., Ashhab, S., & Nori, F. (2014). Quantum simulation. *Reviews of Modern Physics, 86*(1), 153.
68. Lloyd, S. (1996). Universal quantum simulators. *Science, 273*(5278), 1073–1078.
69. Bauer, B., Bravyi, S., Motta, M., & Kin-Lic Chan, G. (2020). Quantum algorithms for quantum chemistry and quantum materials science. *Chemical Reviews, 120*(22), 12685–12717.
70. McArdle, S., Endo, S., Aspuru-Guzik, A., Benjamin, S. C., & Yuan, X. (2020). Quantum computational chemistry. *Reviews of Modern Physics, 92*(1), 015003.
71. Cao, Y., Romero, J., & Aspuru-Guzik, A. (2018). Potential of quantum computing for drug discovery. *IBM Journal of Research and Development, 62*(6), 6:1-6:20.
72. Prousalis, K., & Konofaos, N. (2019). Alpha quantum pattern recognition method for improving pairwise sequence alignment. *Science and Reports, 9*(1), 7226.
73. Kapli, P., Yang, Z., & Telford, M. J. (2020). Phylogenetic tree building in the genomic age. *Nature Reviews Genetics, 21*(7), 428–444.
74. Onodera, W., Hara, N., Aoki, S., Asahi, T., & Sawamura, N. (2023). Phylogenetic tree reconstruction via graph cut presented using a quantum-inspired computer. *Molecular Phylogenetics and Evolution, 178*, 107636.

75. Harrow, A. W., Hassidim, A., & Lloyd, S. (2009). Quantum algorithm for linear systems of equations. *Physical Review Letters, 103*(15), 150502.
76. S.K. Leyton, T.J. Osborne, A quantum algorithm to solve nonlinear differential equations, *arXiv preprint* arXiv:0812.4423 *(2008).*
77. Berry, D. W. (2014). High-order quantum algorithm for solving linear differential equations. *Journal of Physics A: Mathematical and Theoretical, 47*(10), 105301.
78. Childs, A. M., & Liu, J.-P. (2020). Quantum spectral methods for differential equations. *Communications in Mathematical Physics, 375*(2), 1427–1457.
79. Dill, K. A., & MacCallum, J. L. (2012). The protein-folding problem, 50 years on. *Science, 338*(6110), 1042–1046.
80. Perdomo, A., Truncik, C., Tubert-Brohman, I., Rose, G., & Aspuru-Guzik, A. (2008). Construction of model Hamiltonians for adiabatic quantum computation and its application to finding low-energy conformations of lattice protein models. *Physical Review A, 78*(1), 012320.
81. Perdomo-Ortiz, A., Dickson, N., Drew-Brook, M., Rose, G., & Aspuru-Guzik, A. (2012). Finding low-energy conformations of lattice protein models by quantum annealing. *Science and Reports, 2*, 571.
82. Li, R. Y., Di Felice, R., Rohs, R., & Lidar, D. A. (2018). Quantum annealing versus classical machine learning applied to a simplified computational biology problem. *npj Quantum Information*. https://doi.org/10.1038/s41534-018-0060-8
83. Weidner, F. M., Schwab, J. D., Wölk, S., Rupprecht, F., Ikonomi, N., Werle, S. D., Hoffmann, S., Kühl, M., & Kestler, H. A. (2023). Leveraging quantum computing for dynamic analyses of logical networks in systems biology. Patterns. https://doi.org/10.1016/j.patter.2023.100705
84. Lau, B., Emani, P. S., Chapman, J., Yao, L., Lam, T., Merrill, P., Warrell, J., Gerstein, M. B., & Lam, H. Y. (2023). Insights from incorporating quantum computing into drug design workflows. *Bioinformatics, 39*(1), 789.
85. Sohail, A., & U. Ashiq U. (2023). Quantum inspired improved AI computing for the sensors of cardiac mechano-biology. *Sensors International, 4*, 100212.
86. Fox, D. M., MacDermaid, C. M., Schreij, A. M., Zwierzyna, M., & Walker, R. C. (2022). RNA folding using quantum computers. *PLOS Computational Biology, 18*(4), e1010032.
87. Schuld, M., & Petruccione, F. (2018). Quantum ensembles of quantum classifiers. *Scientific Reports, 8*(1), 2772.
88. Wang, X., Ma, Y., Hsieh, M.-H., & Yung, M.-H. (2021). Quantum speedup in adaptive boosting of binary classification, Science China Physics. *Mechanics & Astronomy, 64*(2), 220311.
89. S. Arunachalam, R. Maity, Quantum boosting, In: *International Conference on Machine Learning, PMLR*, 2020, pp. 377–387.
90. Lloyd, S., Garnerone, S., & Zanardi, P. (2016). Quantum algorithms for topological and geometric analysis of data. *Nature Communications, 7*(1), 10138.
91. A. Monras, A. Beige, K. Wiesner, Hidden quantum Markov models and non-adaptive read-out of many-body states, *arXiv preprint* arXiv:1002.2337 *(2010).*
92. S. Srinivasan, G. Gordon, B. Boots, Learning hidden quantum Markov models, In: *International Conference on Artificial Intelligence and Statistics, PMLR*, 2018, pp. 1979–1987.
93. S. Lloyd, M. Mohseni, P. Rebentrost, Quantum algorithms for supervised and unsupervised machine learning, *arXiv preprint* arXiv:1307.0411 *(2013).*
94. Wiebe, N., Kapoor, A., & Svore, K. M. (2015). Quantum nearest-neighbor algorithms for machine learning. *Quantum Information and Computation, 15*(3–4), 318–358.
95. I. Kerenidis, J. Landman, A. Luongo, A. Prakash, q-means: A quantum algorithm for unsupervised machine learning, Advances in neural information processing systems, *arXiv preprint* arXiv:2006.02388 *(2019).*
96. Zhao, Z., Pozas-Kerstjens, A., Rebentrost, P., & Wittek, P. (2019). Bayesian deep learning on a quantum computer. *Quantum Machine Intelligence, 1*, 41–51.
97. M. Denil, N. de Freitas, NIPS Deep Learning and Unsupervised Feature Learning Workshop, NIPS, 2011.
98. V. Dumoulin, I. Goodfellow, A. Courville, Y. Bengio, On the challenges of physical implementations of RBMs, In: *Proceedings of the AAAI Conference on Artificial Intelligence*, 2014.
99. Benedetti, M., Realpe-Gómez, J., Biswas, R., & Perdomo-Ortiz, A. (2016). Estimation of effective temperatures in quantum annealers for sampling applications: A case study with possible applications in deep learning. *Physical Review A, 94*(2), 022308.
100. N. Wiebe, A. Kapoor, K.M. Svore, Quantum deep learning, *arXiv preprint* arXiv:1412.3489 *(2014).*
101. E.R. Anschuetz, Y. Cao, Realizing quantum Boltzmann machines through eigenstate thermalization, *arXiv preprint* arXiv:1903.01359 *(2019).*
102. Zhao, Z., Fitzsimons, J. K., & Fitzsimons, J. F. (2019). Quantum-assisted Gaussian process regression. *Physical Review A, 99*(5), 052331.
103. Zhao, Z., Fitzsimons, J. K., Osborne, M. A., Roberts, S. J., & Fitzsimons, J. F. (2019). Quantum algorithms for training Gaussian processes. *Physical Review A, 100*(1), 012304.
104. Khoshaman, A., Vinci, W., Denis, B., Andriyash, E., Sadeghi, H., & Amin, M. H. (2018). Quantum variational autoencoder. *Quantum Science and Technology, 4*(1), 014001.
105. Rebentrost, P., Mohseni, M., & Lloyd, S. (2014). Quantum support vector machine for big data classification. *Physical Review Letters, 113*(13), 130503.
106. R. Chatterjee, T. Yu, Generalized coherent states, reproducing kernels, and quantum support vector machines, *arXiv preprint* arXiv:1612.03713 *(2016).*
107. Schuld, M., & Killoran, N. (2019). Quantum machine learning in feature hilbert spaces. *Physical Review Letters, 122*(4), 040504.
108. Schuld, M., Sinayskiy, I., & Petruccione, F. (2016). Prediction by linear regression on a quantum computer. *Physical Review A, 94*(2), 022342.
109. Wang, G. (2017). Quantum algorithm for linear regression. *Physical Review A, 96*(1), 012335.
110. Dutta, S., Suau, A., Dutta, S., Roy, S., Behera, B. K., & Panigrahi, P. K. (2020). Quantum circuit design methodology for multiple linear regression. *IET Quantum Communication, 1*(2), 55–61.
111. G. Li, Y. Wang, Y. Luo, Y. Feng, Quantum data fitting algorithm for non-sparse matrices, *arXiv preprint* arXiv:1907.06949 *(2019).*
112. Benedetti, M., Realpe-Gómez, J., Biswas, R., & Perdomo-Ortiz, A. (2017). Quantum-assisted learning of hardware-embedded probabilistic graphical models. *Physical Review X, 7*(4), 041052.
113. Lloyd, S., Mohseni, M., & Rebentrost, P. (2014). Quantum principal component analysis. *Nature Physics, 10*(9), 631–633.
114. Kak, S. C. (1995). Quantum neural computing. *Advances in Imaging and Electron Physics, 94*, 259–313.
115. Zak, M., & Williams, C. P. (1998). Quantum neural nets. *International Journal of Theoretical Physics, 37*(2), 651–684.
116. Y. Cao, G.G. Guerreschi, A. Aspuru-Guzik, Quantum neuron: an elementary building block for machine learning on quantum computers, *arXiv preprint* arXiv:1711.11240 *(2017).*
117. Low, G. H., Yoder, T. J., & Chuang, I. L. (2014). Quantum inference on Bayesian networks. *Physical Review A, 89*(6), 062315.
118. N. Wiebe, C. Granade, Can small quantum systems learn?, *arXiv preprint* arXiv:1512.03145 *(2015).*

119. Kerenidis, I., Luongo, A., & Prakash, A. (2020). Quantum expectation-maximization for Gaussian mixture models. *International Conference on Machine Learning, PMLR, 119*, 5187–5197.
120. Miyahara, H., Aihara, K., & Lechner, W. (2020). Quantum expectation-maximization algorithm. *Physical Review A, 101*(1), 012326.
121. Lloyd, S., & Weedbrook, C. (2018). Quantum generative adversarial learning. *Physical Review Letters, 121*(4), 040502.
122. Dallaire-Demers, P.-L., & Killoran, N. (2018). Quantum generative adversarial networks. *Physical Review A, 98*(1), 012324.
123. Dunjko, V., Taylor, J. M., & Briegel, H. J. (2016). Quantum-enhanced machine learning. *Physical Review Letters, 117*(13), 130501.
124. Cong, I., Choi, S., & Lukin, M. D. (2019). Quantum convolutional neural networks. *Nature Physics, 15*(12), 1273–1278.
125. Hollenberg, L. C. (2000). Fast quantum search algorithms in protein sequence comparisons: quantum bioinformatics. *Physical Review E, 62*(5), 7532.
126. Abdesslem, L., Soham, M., & Mohamed, B. (2006). Multiple sequence alignment by quantum genetic algorithm. *Sat, 2*, 1.
127. Prousalis, K., & Konofaos, N. (2019). A quantum pattern recognition method for improving pairwise sequence alignment. *Scientific Reports, 9*(1), 7226.
128. A. Sarkar, Z. Al-Ars, C.G. Almudever, K. Bertels, An algorithm for DNA read alignment on quantum accelerators, *arXiv preprint* arXiv:1909.05563 *(2019).*
129. Niroula, P., & Nam, Y. (2021). A quantum algorithm for string matching. *npj Quantum Information, 7*(1), 37.
130. D. Ellinas, P. Jarvis, Quantum simulation of phylogenetic trees, *arXiv preprint* arXiv:1105.1582 *(2011).*
131. Banchi, L., Fingerhuth, M., Babej, T., Ing, C., & Arrazola, J. M. (2020). Molecular docking with Gaussian boson sampling. *Science Advances*. https://doi.org/10.1126/sciadv.aax1950
132. B. Kerstin, B. Dmytro, F. Terry, O. Tobias, S. Robert, W. Ramona, Efficient learning for deep quantum neural networks, Nature (2019).
133. Killoran, N., Bromley, T. R., Arrazola, J. M., Schuld, M., Quesada, N., & Lloyd, S. (2019). Continuous-variable quantum neural networks. *Physical Review Research, 1*(3), 033063.
134. Abbas, A., Sutter, D., Zoufal, C., Lucchi, A., Figalli, A., & Woerner, S. (2021). The power of quantum neural networks. *Nature Computational Science, 1*(6), 403–409.
135. J. Jiang, X. Zhang, C. Li, Y. Zhao, R. Li, Generalization study of quantum neural network, *arXiv preprint* arXiv:2006.02388 *(2020).*
136. P. Rebentrost, T.R. Bromley, C. Weedbrook, S. Lloyd, A quantum recurrent neural network, *arXiv preprint* arXiv:1710.03599 *(2017).*
137. Alghassi, H., Dridi, R., Robertson, A. G., & Tayur, S. (2019). Quantum and quantum-inspired methods for de novo discovery of altered cancer pathways. *bioRxiv, 6*(7), 845719.
138. R. Balu, A. Borle, Bayesian networks based hybrid quantum-classical machine learning approach to elucidate gene regulatory pathways, *arXiv preprint* arXiv:1901.10557 *(2019).*
139. Negre, C. F., Ushijima-Mwesigwa, H., & Mniszewski, S. M. (2020). Detecting multiple communities using quantum annealing on the D-Wave system. *PLoS ONE, 15*(2), e0227538.
140. Perdomo-Ortiz, A., Dickson, N., Drew-Brook, M., Rose, G., & Aspuru-Guzik, A. (2012). Finding low-energy conformations of lattice protein models by quantum annealing. *Scientific Reports, 2*(1), 1–7.
141. T. Babej, M. Fingerhuth, Coarse-grained lattice protein folding on a quantum annealer, *arXiv preprint* arXiv:1811.00713 *(2018).*
142. Govenius, J., Lake, R., Tan, K., & Möttönen, M. (2016). Detection of zeptojoule microwave pulses using electrothermal feedback in proximity-induced Josephson junctions. *Physical Review Letters, 117*(3), 030802.
143. Montanaro, A. (2017). Quantum pattern matching fast on average. *Algorithmica, 77*, 16–39.
144. Ramesh, H., & Vinay, V. (2003). String matching in O (n+ m) quantum time. *Journal of Discrete Algorithms, 1*(1), 103–110.
145. Li, R. Y., Di Felice, R., Rohs, R., & Lidar, D. A. (2018). Quantum annealing versus classical machine learning applied to a simplified computational biology problem. *npj Quantum Information, 4*(1), 14.
146. Sarkar, A., Al-Ars, Z., & Bertels, K. (2021). QuASeR: Quantum Accelerated de novo DNA sequence reconstruction. *PLoS ONE, 16*(4), e0249850.
147. Boev, A., Rakitko, A., Usmanov, S., Kobzeva, A., Popov, I., Ilinsky, V., Kiktenko, E., & Fedorov, A. (2021). Genome assembly using quantum and quantum-inspired annealing. *Scientific Reports, 11*(1), 13183.