A Quantum Algorithm for Pairwise Sequence Alignment

by

Md.Rabiul Islam Khan
22341057

Shaikh Farhan Rafid
20241034

Raya Jahan
18241012

A thesis submitted to the Department of Computer Science and Engineering
in partial fulfillment of the requirements for the degree of
B.Sc. in Computer Science

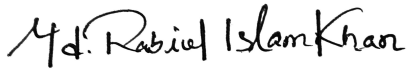
Department of Computer Science and Engineering
Brac University
January 2023

Declaration

It is hereby declared that

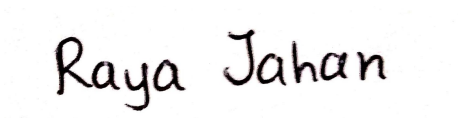
1. The thesis submitted is my/our own original work while completing degree at Brac University.
2. The thesis does not contain material previously published or written by a third party, except where this is appropriately cited through full and accurate referencing.
3. The thesis does not contain material which has been accepted, or submitted, for any other degree or diploma at a university or other institution.
4. We have acknowledged all main sources of help.

Student's Full Name & Signature:



Md.Rabiul Islam Khan
22341057

Shaikh Farhan Rafid
20241034



Raya Jahan
18241012

Approval

The thesis titled "A Quantum Algorithm for Pairwise Sequence Alignment" submitted by

1. Md.Rabiul Islam Khan (22341057)
2. Shaikh Farhan Rafid (20241034)
3. Raya Jahan (18241012)

of Fall, 2022 has been accepted as satisfactory in partial fulfillment of the requirement for the degree of B.Sc. in Computer Science on January 19, 2023.

Examining Committee:

Supervisor:
(Member)

Shadman Shahriar
Lecturer
Department of Computer Science and Engineering
Brac University

Co-Supervisor:
(Member)

Farhan Feroz
Lecturer
Department of Computer Science and Engineering
Brac University

Program Coordinator:
(Member)

Md. Golam Rabiul Alam, PhD
Associate Professor
Department of Computer Science and Engineering
Brac University

Head of Department:
(Chair)

Sadia Hamid Kazi, PhD
Chairperson and Associate Professor
Department of Computer Science and Engineering
Brac University

Abstract

DNA Sequence Alignment is the process of aligning DNA sequences in order to identify similarities between multiple sequences. The primary reason is to sort out the evolutionary, functional or structural relationships within sequences. In our thesis, we are trying to design a Quantum Algorithm for finding the optimal alignments between DNA sequences. This particular method of finding DNA sequences works by mapping the sequence alignment problem to a path searching problem through a 2D graph. The transition which converges to a fixed path on the graph is based on a proposed oracle for profit calculation. In order to generate a graph and create random paths through it, we searched and analyzed several Quantum Walk Algorithms to align sequences in contrast to classical random walk search algorithms. Our primary goal is to align DNA sequences and find the optimal alignments in a faster and efficient way with Grover's search algorithm. It is capable of quadratic speeding up of any unstructured search problem. The intent here is to provide a comprehensive elaboration of how path searching algorithm works on DNA sequences according to a quantum algorithm.

Keywords: Quantum Computing; Qubit; Superposition; Entanglement; Quantum Logic Gate; Grover's Search Algorithm; Quantum Fourier Transform; Quantum Random Access Memory; DNA Sequence; Sequence Alignment

Acknowledgement

We are highly indebted to our advisors Mr. Shadman Shahriar and Mr. Farhan Feroz of the Department of Computer Science and Engineering, BRAC University. This thesis work would never be possible without the guidance and constant supervision of our mentor, Mr. Shadman Shahriar. We found him holding our back whenever we were in dire need. His fine knowledge and technical skills immensely helped us in establishing this thesis. Whenever we needed the spark, he was the one to ignite us. We are truly grateful to Mr. Shadman for his time and effort. Lastly, we are also grateful to Md. Shahnewaz Ahmed and Mr. Sowmitra Das for founding the basis of Quantum Computing research in this university.

Table of Contents

List of Figures

List of Tables

Nomenclature

The next list describes several symbols & abbreviation that will be later used within the body of the document

DNA Deoxyribonucleic Acid

QFT Quantum Fourier Transform

qRAM Quantum Random Access Memory

RNA Ribonucleic Acid

..

Chapter 1

Introduction

Since the midst of the last century, there has been an issue to be addressed regarding classical computers, that are not sustainable to keep up the pace with emerging necessities of speeding up the processing of information and rapid shrink in size of the computer chip which may break the physical barrier. Due to these limitations, quantum mechanics is being considered as a powerful counterpart in the race of a future automaton to cope up with the challenges [1].
The alignment of sequences; arranging sequences of DNA, RNA or Proteins to demonstrate regions of similarity in the biological, structural and identical connections requires adequate amount of information and time [2]. With the progress of quantum computation in this decade, the sequence alignment problem may be solved with nearly full precision in contrast to previous probabilistic approaches.
Quantum mechanics harnesses the phenomenon of superposition, entanglement, tunneling and annealing to solve problems that take a tremendous amount of time [3]. Superposition allows quantum bits to be represented with 0, 1 or both at the same time [4]. If two systems are strongly co-related to each other then gaining the information of one system will immediately provide the information for the other system through the effect is quantum entanglement [5].

1.1 Problem Statement

Finding the optimal alignments of DNA sequences has been one of the most challenging aspects of Bio informatics. To find the optimal alignment, there have been several computational approaches for both pairwise and multiple/global sequences. Although several computational algorithms have been implemented to solve the sequence alignment problem, none of these approaches guarantee to sort out the optimal alignment in the most efficient way.
These approaches include the naive approaches, slow yet working dynamic programming approach and more efficient approaches for large databases such as heuristic and probabilistic approaches.
With the progression of Quantum computers within this decade, the applications of these computing machines are being tested theoretically. Particularly in bio informatics, quantum algorithms are believed to solve some of the most complex computational problems. Although DNA sequence alignment is a decade long computational challenge and has been proven to be useful in many applications of bio informatics, there has not been any significant approaches to develop a suitable Quantum algorithm in order to

find the optimal alignment.
The most naive approach is to search the similarities between two sequences i.e. pairwise alignment. Needleman and Wunsch (1970) presented the inaugural approach with dynamic programming which was meant for protein sequences [6]. However, because of the running time and memory requirements, the dynamic approaches are proven to be quite impractical.
Moreover, although heuristic and stochastic methods are more efficient, none of these algorithms properly guarantee to sort out the optimal alignment. Also, running time and memory requirements also have been an issue in case of the classical algorithmic approaches.
In our thesis, we look forward to harness the computational capabilities of quantum computers to solve the problem of aligning sequences. Rather than taking a probabilistic approach that is quite practical for classical computing, we propose a deterministic approach and provide a proper guarantee to figure out the optimal path. We can also be hopeful to speed up the time required for figuring out a desired solution.

1.2 Research Objectives

The primary goal is to figure out the approach to build a proper quantum walk algorithm and generate a graph for multiple DNA sequences. Moreover, another important task will be to sort out the best possible way to define 'path score' for the problem statement. The focus is to implement a deterministic approach to figure out the optimal solution. In order to do so, getting familiar with path search techniques thorough a graph is a necessity.
Figuring out an approach to define path cost will be a major task. The optimal path should correspond to the defined path score.

In summary, our purpose in this thesis is to build and implement a proper Quantum Algorithm to solve sequence alignment problems in a deterministic manner.

The intention is to address the followings and recognize the best possible answers,

1. Build a proper two-dimensional graph for the problem statement.
2. Generate all possible paths through the graph on the basis of proposed path cost and create best chances to find out the optimal solution.
3. Figure out a potential quantum algorithm that will be a proper selection for the implementation of the DNA sequence alignment problem.
4. Improve the time and space complexity as well as increase the probability to find out the optimal alignment. Compare the proposed approach with its classical counterparts.
5. Measure time and memory requirements and analyze the benefits of quantum computation in DNA sequence alignment problems.

Chapter 2

Quantum Computing

2.1 Quantum Computing

A comprehensive study of quantum computing is provided here [4]. Quantum computers are information processing and computational machines that take advantages of quantum theory phenomena. It can be immensely helpful for certain tasks where the computing machines can considerably outperform even the most powerful supercomputers. Conventional computers, store data using "bits", which may be either 0s or 1s. On the other hand, qubit (quantum bit), which is the core/fundamental component of a quantum computer can be 0s or 1s at the same time by the property known as superposition. Although, after a qubit is measured it is either 0s or 1s. The initial state of the qubit is presented with the probabilistic outcomes of the basis states.
To create a physical qubit, systems such as the spin of an electron or particle of light, photon can be used which hold the properties and aspects of quantum mechanics. Quantum superposition allows these systems to be in several states (configuration) at the same time. Through the phenomenon of Quantum entanglement, qubits are inextricably connected to each other. As a result, a set of qubits can represent numerous states at the same moment.

2.2 Qubit

The functionalities and mechanisms for processing and storing data differ when it comes to quantum and classical computers. As we know, classical computers deal with bits; a bit represents information in a two dimensional classical system, whose states are expressed by 0 and 1.
A qubit (quantum bit) is the quantum equivalent of classical bits. A qubit can be understood by a two state system. We can demonstrate the system with the idea of classical bits. In quantum computation, qubit is the basic unit of computation. Qubits can be represented with a two-by-one matrix where qubits are two dimensional matrices with entries that are complex numbers.
In quantum computing, we represent a qubit which is the unit of information, with "Braket" notation; 0 as $|0\rangle$ and 1 as $|1\rangle$; these are the computational basis of qubits.

$$|0\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \tag{2.1}$$

$$|1\rangle = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \tag{2.2}$$

2.2.1 Quantum Superposition

Superposition plays the role in distinguishing between classical bits and qubits that can be more than just a single system, a linear combination of states.
If we take a random qubit state and the entries are two complex numbers c_0 and c_1,

$$|\psi\rangle = \begin{bmatrix} c_0 \\ c_1 \end{bmatrix} \tag{2.3}$$

Here, $|\psi\rangle = c_0 |0\rangle + c_1 |1\rangle$. Each of the systems should be a normalized vector,

$$||c_0||^2 + ||c_1||^2 = 1 \tag{2.4}$$

In the similar way, n qubits are represented by a superpositional state vector in a 2^n dimensional Hilbert space. A qubit can also be geometrically represented with Bloch Sphere. Here, all the states are presented as projections where the basis states are orthogonal projections to each other.

2.2.2 Bloch Sphere

The pure/actual state of a qubit can be geometrically represented with a "Bloch Sphere"; a representation of all possible states of any two-level quantum mechanical system. For a random qubit with superposition states $|\psi\rangle = a|0\rangle + b|1\rangle$,

$$a = \cos\left(\frac{\theta}{2}\right) \qquad b = e^{i\phi} \sin\left(\frac{\theta}{2}\right)$$

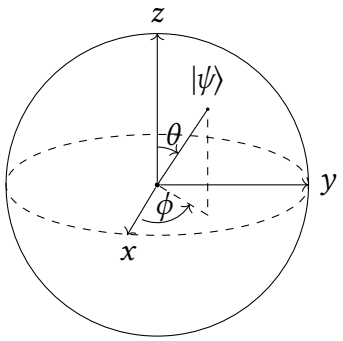


Figure 2.1: Bloch Sphere

2.2.3 Quantum Entanglement

Quantum entanglement is a special kind of state which can't be expressed independently with respect to other states. Quantum entanglement is considered as the most fascinating aspect of quantum computation [5]. In other notation, entangled states are not tensor products of pure states which means these are not separable. In a combined system of two qubits, entanglement provides a crucial role in quantum computation. Example of entangled state can be,

$$\frac{1}{\sqrt{2}} |00\rangle + \frac{1}{\sqrt{2}} |11\rangle$$

2.3 Quantum Logic Gate

Quantum logic gates are reversible, unitary operators and can be described with unitary matrices. Quantum logic gates operate and manipulate on qubit systems. These are the basic quantum circuits. Quantum logic gates are building blocks like their classical counterparts, digital logic gates.
As, quantum logic gates are unitary operators, these gates preserve the inner product. Unitary operators are described as unitary matrices, i.e., the inverse is same as the conjugate transpose.

2.3.1 Pauli Gates (X, Y, Z)

The Pauli gates are three Pauli matrices that are unitary, involutory and hermitian. These gates operate on a single qubit.

$$X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \qquad \text{—}\boxed{X}\text{—} \tag{2.5}$$

$$Y = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix} \qquad \text{—}\boxed{Y}\text{—} \tag{2.6}$$

$$Z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \qquad \text{—}\boxed{Z}\text{—} \tag{2.7}$$

Pauli-X gate is known as a bit-flip as it acts like a classical NOT-gate. Pauli-Z gate performs as a phase-flip gate. Pauli-Y maps $|0\rangle$ to $i|1\rangle$ and $|1\rangle$ to $-i|0\rangle$.

2.3.2 Hadamard Gate

Hadamard gate is one of the most used quantum logic gates. Operating on a single qubit, it can transform the qubit into a superposition of states. In case of computational basis, Hadamard gate can create equal superposition states that are the orthogonal basis of qubits; $|+\rangle$ and $|-\rangle$.

$$H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \qquad \text{—}\boxed{H}\text{—} \tag{2.8}$$

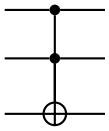
2.3.3 Controlled NOT (CX) gate

$$CNOT = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \tag{2.9}$$

It is a two-qubit gate, one qubit acts as a control bit and the other qubit is the target bit. If the controlled qubit is $|1\rangle$, then the target qubit shall be flipped i.e., Pauli-X applied on the qubit.

2.3.4 Toffoli (CCX) Gate

$$CNOT = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \qquad (2.10)$$



Toffoli gate is three-qubit gate where first two qubits are control bits and the last qubit is the target bit. If the first two qubits are in the state $|1\rangle$, then, Pauli-X is applied on the third qubit.

2.3.5 Rotational Gates

A qubit can be rotated by an angle θ in radians along the x-axis, y-axis, and z-axis which can be called R_x, R_y, R_z respectively using rotational gates. These gates function on the edge of the Bloch Sphere. The matrices of R_x, R_y and R_z can be represented as below:

$$R_x(\theta) = \begin{bmatrix} cos\theta & -isin\theta \\ -isin\theta & cos\theta \end{bmatrix} \qquad (2.11)$$

$$R_y(\theta) = \begin{bmatrix} cos\theta & -sin\theta \\ sin\theta & cos\theta \end{bmatrix} \qquad (2.12)$$

$$R_z(\theta) = \begin{bmatrix} e^{-i\frac{\theta}{2}} & 0 \\ 0 & e^{i\frac{\theta}{2}} \end{bmatrix} \qquad (2.13)$$

Furthermore, R_x, R_y and R_z can be extended to 2-qubit rotational gates which are R_{xx}, R_{yy} and R_{zz} respectively. Along the x-axis, R_{xx} rotate two qubits by θ in radians which is same for other two operators R_{yy} and R_{zz} along with y-axis and z-axis accordingly. The matrices of R_{xx}, R_{yy} and R_{zz} can be represented as below:

$$R_{xx}(\theta) = \begin{bmatrix} cos\theta & 0 & 0 & -isin\theta \\ 0 & cos\theta & -isin\theta & 0 \\ 0 & -isin\theta & cos\theta & 0 \\ -isin\theta & 0 & 0 & cos\theta \end{bmatrix} \qquad (2.14)$$

$$R_{yy}(\theta) = \begin{bmatrix} cos\theta & 0 & 0 & isin\theta \\ 0 & cos\theta & -isin\theta & 0 \\ 0 & -isin\theta & cos\theta & 0 \\ isin\theta & 0 & 0 & cos\theta \end{bmatrix} \qquad (2.15)$$

$$R_{zz}(\theta) = \begin{bmatrix} e^{-i\theta} & 0 & 0 & 0 \\ 0 & e^{i\theta} & 0 & 0 \\ 0 & 0 & e^{i\theta} & 0 \\ 0 & 0 & 0 & e^{-i\theta} \end{bmatrix} \qquad (2.16)$$

2.3.6 Quantum Fourier Transform

Quantum Fourier transform (QFT) is a linear transformation on qubits. It is the quantum analogue of Discrete Fourier transform. Quantum Fourier transform can be represented as a unitary matrix. For each $\omega = \omega_N$,

$$F_N = \frac{1}{\sqrt{N}} \begin{bmatrix} 1 & 1 & 1 & \dots & 1 \\ 1 & \omega & \omega^2 & \dots & \omega^{N-1} \\ 1 & \omega^2 & \omega^4 & \dots & \omega^{2(N-1)} \\ 1 & \omega^3 & \omega^6 & \dots & \omega^{3(N-1)} \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & \omega^{N-1} & \omega^{2(N-1)} & \dots & \omega^{(N-1)(N-1)} \end{bmatrix} \tag{2.17}$$

2.4 Grover's Algorithm

Grover's algorithm is an unstructured searching algorithm to find out an element from an unsorted array of length N in $O(\sqrt{N})$ steps in contrast to classical algorithms which require O(N) evaluations [7].

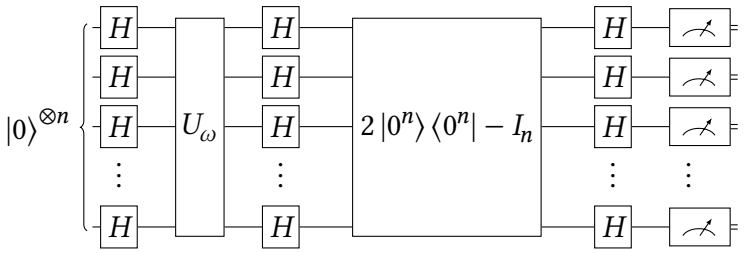


Figure 2.2: Circuit of Grover's Algorithm

The steps for Grover's algorithm:

1. Apply the Hadamard Transform $H^{\otimes n}$;

$$|\psi\rangle = \frac{1}{\sqrt{N}} \sum_{x=0}^{N-1} |x\rangle \tag{2.18}$$

   $|x\rangle$ corresponds to the index register. Initialize the entire system into a uniform superposition of states.

2. Apply the phase inversion oracle i.e., a unitary operator such that,

$$U_w |x\rangle = (-1)^{f(x)} |x\rangle \tag{2.19}$$

   Here, f(x) is the oracle function. f(x) = 1, when the data pointed by x is an item that has been sought for.

3. Apply the Grover diffusion operator;
   - Hadamard Transform $H^{\otimes n}$
   - Conditional phase shift operation, 2|0⟩⟨0| - I
   - Hadamard Transform $H^{\otimes n}$

4. Measure in the computational basis.

A slightly modified Grover's search algorithm has to be implemented in this work to find the maximum element only.

Chapter 3

Feynman Path Integral

3.1 Classical Mechanics

The study of how bodies move when they are subject to external forces is what 'Classical Mechanics' deals with. Any force that can modify an object's velocity, e.g: Sir Isaac Newton described everyday events using his three laws of motion. The laws generally describe the movement of anything, from a galaxy to a little particle. Those physical laws, actually their mathematical representations are verified by experimentation, form the foundation of mechanics [8]. The modern computer is based on the classical binary bit, which are represented as 0 and 1. These bits are used as the single most fundamental element of the computer. These bits make up the whole system of hardware, based on the on-and-off logical operation in tiny chips in processors. These logical operations are done in logical gates, such as OR, AND, XOR and XNOR. These gates have their equivalent devices in quantum computing as well.

3.1.1 Action

Action is a method of finding equations of motions of any given physical system in the classical domain of physics. It is defined in many ways. One of them is the definition of what is known as a "Functional". These functional's paved the abstract way-generating path.

$$S[q(t)] = \int_{t_1}^{t_2} L\left(q(t), \dot{q}(t), t\right) dt$$

Here S is a functional that takes an input function $q(t)$ of any system in-between time t_1 and t_2. Here q denotes the generalized coordinates. The action here is the integral of Lagrangian L as per the evolution from t_1 to t_2, which takes $q(t)$ and $\dot{q}(t)$.

3.1.2 Principle of Least Action

In everyday life, nature tends to minimize the energy of any system, or take the path of least energy fluctuation. Such as the spherical shape of a bubble makes the surface tension have the lowest potential; also when a light ray passes through two mediums of varying density, it takes a not-so-straight path which ensures that is the optimal path. With infinite such mediums, one can approximate what is known as the Brachistochrone

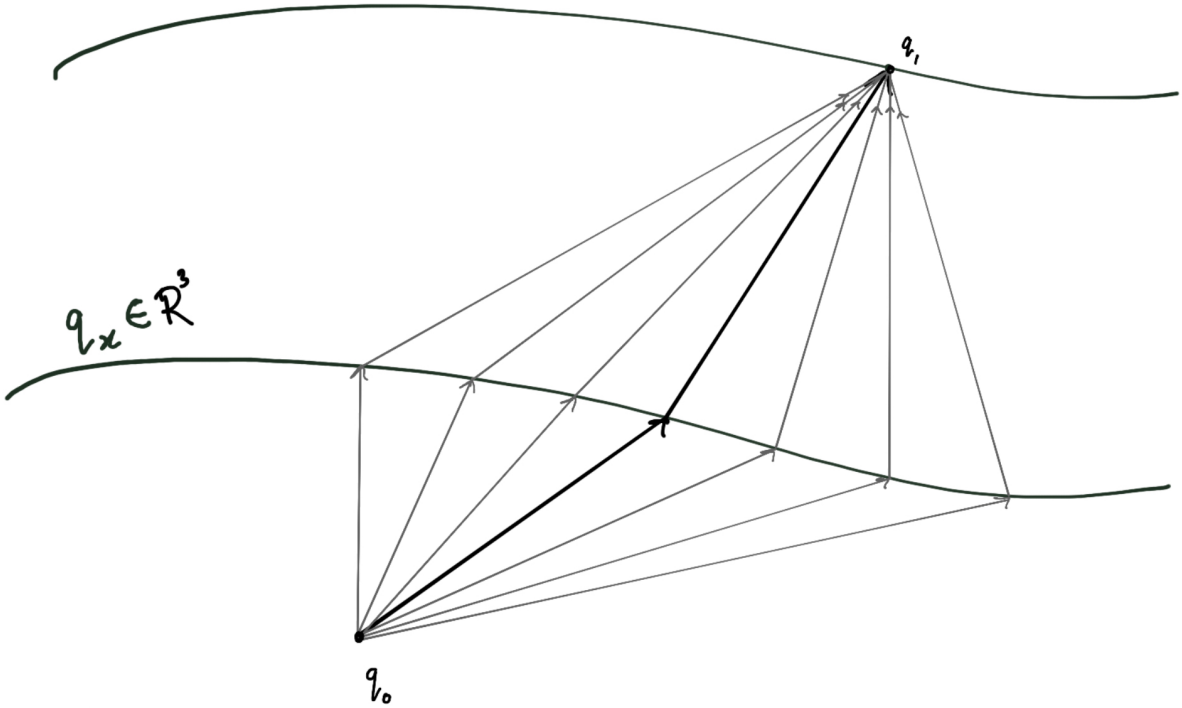
curve. This nature of moving bodies willing to take the most optimal path is known as The Principle of Least Action.
On any system, let K be the Kinetic energy and V be the Potential energy. Lagrangian L can be defined as $(K - V)$. If the time elapses and the system moves from t_0 to t_1, the Lagrangian L can be integrated in that time with the integration measure dt, which will formulate the principle of least action. Here, action is represented by S. [9]

$$S = \int_{t_0}^{t_1} (K - V)dt$$

$$S = \int_{t_0}^{t_1} Ldt$$

These integrals generate a path, one of the examples was given with light rays passing through a density gradient. Mainly, these paths are continuous in nature, and thus lie in the realms of continuous Quantum Mechanics.

3.2 Path Integral Formulation



Path integral formulation of quantum mechanics revolutionized the field of Physics. Richard Feynman developed the mathematical tool in 1948 using the Principle of Least Action of Classical Mechanics. The formulation of Path Integral removes the unique single path notion of a path. It brings in a weighted sum of infinite possible paths, which in the end gives the trajectory of a particle in a generalized coordinate world. Due to the superposition nature of the quantum wave function, the "particle" can be formulated in all possible paths from point A to B in time t_a to t_b. Here t_a and t_b are the 2 endpoints of the sliced time sequence.

$$t_a = t_0 < t_1 < .... < t_{n-1} < t_n < t_{n+1} = t_b$$

Similarly to Classical Mechanics, the Lagrangian here is in exponential form due to the formulation of the wave function with necessary constants such as i and $\hbar$, and it is integrated with respect to dx_0, dx_1,...,dx_n from $-\infty$ to $+\infty$.

$$\int_{-\infty}^{+\infty} ... \int_{-\infty}^{+\infty} exp\left(\frac{i}{\hbar}\int_{t_a}^{t_b} L(x(t), v(t))dt\right) dx_0...dx_n$$

The arguments of the Lagrangian L are position variable $x(t)$ and velocity $v = \dot{x}(t)$. This time integration is approximated by the totality of n terms. When n tends to infinity, it becomes a functional [10].

3.3 Analogy of Path Integral in DNA Sequencing

The reason to demonstrate Richard Feynman's path integral Formalism is that we will be using quantum computation in regard to finding the optimal path that will sequence genomes at an exponential rate. We will explain this in greater detail, however, to say briefly, a quantum computer uses the Physics of Superposition of wave functions to calculate many possible sequencing at the same time.
When two DNA sequences are aligned in an orthogonal form into a 2D grid, where the selection of the path will determine the optimal sequence between the two genes; classical algorithms explore each possible path one by one to find the approximation of the best possible path, or the local maxima of the hill. But as Richard Feynman formulated in Path Integral, a particle traversing from point A to point B will take all the possible paths at the same time; the quantum algorithm will also similarly traverse all the possible paths at the same time from the beginning to the end. This process will not find the approximated, but the most optimal path based on some certain heuristic function to calculate the path cost. With the allegory of the Hill Climb algorithm, if we use the quantum approach, we can find the global maxima of the hill.
Another analogy can be made with the quantum algorithm which generates a path in a discrete manner, which looks like a maze, where the traveler is starting from a position and he has to find the way out of the maze by choosing the only path that exists. The only path in this context is the path of highest profit, and the quantum nature can be used in this situation to traverse all the paths of the maze at the same time and find the way out in a nearly extremely efficient manner.

Chapter 4

DNA Sequence Alignment

4.1 DNA Sequence

DNA sequences are sequences of nucleic acid that represent the order of the nucleotides. The sequences of nucleic acid (DNA) consists of four characters or letters [11]. These letters are four nucleotide bases, A, G, C and T representing Adenine, Guanine, Cytosine and Thymine respectively. The necessities of the knowledge of DNA sequences vary from biological discoveries to medical researches and to find out a faster method to align sequences in order to accelerate bio-informatics research [12].

4.2 Sequence Alignment

Sequence alignment is the technique of analyzing and uncovering similarities between biological sequences; in a variety of bio-informatics applications, sequence alignment technique is used to align sequences of DNA, RNA, proteins and non-biological sequences as well [13]. The goal is to find an optimal sequence by calculating distance cost between sequences.
Typically, when dealing with alignment problems using dynamic approaches, calculating edit/path distance between sequences, scores are assigned on the basis of distances. The similarities between sequences are calculated with the assigned score which is defined on the basis of a proposed method. The most general approach is to assign the score '1' in case of the same characters in the same index and assign '-1' otherwise [6]. 'Hamming Distance' is an approach to find the non-matching characters. For the following example, the calculated hamming distance is 2.

A T T C G G A
A T T G C G A

Another approach to find path score is 'Edit Distance' that calculates the minimum number of operations required to change one sequence to another so that the sequences completely match each other. For example,

A T G A C C G T
A C G T C C G T

Here, the calculated edit distance is 2.

4.2.1 Global, Local and Pairwise

The alignments can be pairwise, local or global. Global alignment algorithms are typically implemented to match sequences when a pair of sequences is entirely quite similar [14]. Local alignment algorithms are useful when there are regions/local portions of similarity [15]. Several dynamic approaches have been initiated to align global and pairwise sequences. Pairwise alignment approaches are applied to align two query sequences whether these are local or global. Although, there are three methods for the technique, word method, dynamic programming and dot matrix; we have extensively focused on the dynamic approaches in our work [13].

4.2.2 Substitution Matrix

Substitution matrices are indicator of the frequency of the evolutionary changes in the genomics structures i.e., nucleotide and protein sequences. In the context of sequence alignment, these stochastic matrices are used to calculate identical similarities between sequences [16].

4.2.3 Multiple Sequence Alignment

The extended approach to find out optimal sequence(s) from multiple query sequences; termed as multiple sequence alignment. The goal is to incorporate more sequences at a time rather than a couple of sequences at a time. Primary reasons are due to the hypothetical assumptions, when two biological bodies are thought to be connected through an evolutionary change. These approaches are typically computationally complex; leading to NP-hard optimization problems on frequent occasions [17].

Chapter 5

Literature Review

The motivation of our work has been initiated from Keith and Kroese (2002) who proposed a stochastic method to find the optimal alignment of DNA sequences [18]. The approach is to create random paths through a graph. Each of the generated paths are assigned with scores and are used to update the transition probabilities of the Markov chain. When this process converges into a path it will correspond to an optimal or sub-optimal sequence. The update of the transition probabilities is based on the Cross Entropy Method. The minimization of Cross Entropy distance between two distributions dynamically updates the transition probabilities.
Although, the primary intention was to implement a stochastic/probabilistic approach with quantum walk algorithms, we have modified our work to be deterministic. The goal is to find the maximum element/profit from the generated paths. We have referred to Ahuja and Kapoor (1999) who demonstrated a modified version of Grover's search algorithm to find the maximum element only [19]. Grover's algorithm, can quadratically speed up the unstructured search problems with $O(\sqrt{N})$ for an array of length N, was introduced in 1996 by Lev Grover [7].

5.1 Works on Sequence Alignment

DNA sequence alignment algorithms have been built and implemented to both pairwise and global alignments and has been a topic of extensive studies. The algorithms can be classified as deterministic, stochastic and heuristic. Needleman and Wunsch (1970) presented a dynamic approach for global alignments to sort out protein sequences which was the first approach to find sequence alignments [6]. Similar approach for local alignments have been initiated by Smith and Waterman (1981) [20]. The dynamic algorithms can guarantee to find optimal paths on the condition of defining a good scoring function which is quite unsuitable for larger sequences. These algorithms create a matrix, where each cell represents the similarity score of the sub-string of the first sequence ending at that row and the sub-string of the second sequence ending at that column. The algorithm then fills in the matrix by comparing each residue of the two sequences and scoring their similarity.
The Gibbs sampler approach proposed by Lawrence, Altschul, Boguski, Liu, Neuwald, and Wootton (1993) presented a stochastic approach [21]. Stochastic approach works better than dynamic approach for larger datasets but there remains the probability of returning an optimal or suboptimal path. Although, we are primarily focusing on deterministic approaches to address our problem statement. Stochastic approaches are quite

similar to heuristic approaches.
Heuristic approach may also return a suboptimal path. These algorithms define problem specific search techniques in contrast to the stochastic approaches. Pevzner (1992) presented some examples and spotted the differences between the dynamic and heuristic approaches [22].

5.2 Quantum Walk Algorithm

To understand how fundamental algorithms such as Grover's algorithm can be implemented on path searching problems as well as on graphs, we have gone through several articles on quantum walk algorithms. Venegas-Andraca (2012), presented the theoretical foundations as well as the advancement of discrete and continuous quantum walks in his comprehensive review of quantum walk algorithms [23].
Lovett, Cooper, Everitt, Trevers and Kendon (2010) demonstrated that along with continuous quantum walk, discrete quantum walks are universal for quantum computation as discrete quantum walk is able to implement universal gate sets [24]. Moreover, their proposed set of components provide perfect state transfer for discrete time quantum walk.
To get an overview of quantum walk on graphs, we have referred to Aharonov, Ambainis, Kempe and Vazirani (2001) [25]. They generalized random walk on finite graphs for various aspects of quantum computation.

5.3 Recent Works

Sanchez, Salami, Ramirez and Valero (2006) presented a micro-architecture performance analysis of recognized biological applications in order to compare and align sequences [26]. They adopted a methodology based on simulation and performed detailed workload characterization of the applications regarding sequence comparison as well as the alignment task.
Other works include, Tulsi (2008) demonstrating a quantum algorithm to find an item from N items in $O(\sqrt{N \ln N})$ steps in contrast to previous walk algorithms with $O(\sqrt{N} \ln N)$ complexity [27]. The proposed improvement is $O(\sqrt{\ln N})$ and achieved by controlling quantum walk on lattices using an ancilla qubit.
Dürr et all. (2006) [28], provided tight lower and upper bounds for bounded error quantum query complexity of connectivity, strong connectivity, single source shortest path and minimum spanning tree problems. They showed that for a matrix model the quantum query complexity of a single source shortest path problem is $O(n^{3/2} \log^2 n)$ for an array model the complexity turns out to be $O(\sqrt{nm} \log^2 n)$.
Moreover, in the survey paper of Santha (2008), intuitive treatment has been given of the discrete time quantization of classical Markov chains and stated that, Grover's search algorithm along with other quantum walk algorithms by Ambainis et al. can be the quantum analogues of classical search problems [29]. We relied on Ambainis (2003) to get an overview of quantum walks and the potential applications [30].

5.3.1 Quantum Algorithms for Aligning Sequences

With the progression of a fully performing quantum computer, the sequence alignment problem is being addressed over the last couple of years. A pattern matching algorithm based on the hamming distance named *QiBAM* has been proposed by Sarkar et al.(2019) which can provide quadratic speedup and can be implemented using Grover's algorithm [31].
Quite different from the previous approaches, an updated approach of connecting dot-matrix plotting and quantum pattern recognition has been initiated by Prousalis (2019) in order to improve the process of aligning sequences [32].

Chapter 6

Framework

In order to address the sequence alignment problem, this work is focusing on finding out the optimal sequence of any pairwise alignment; dealing with a pair of sequences in contrast to global alignment which deals with a large number of sequences in a database. The proposed method starts with building a proper graph. For a pair of sequences, a 2-Dimensional graph is required to be implemented. The method may refer to path searching algorithms with the basis of calculated path cost/profit.

6.1 2D Graph

The method to build a proper graph has been initiated from Needleman and Wunsch (2002) [6]. A generated edit graph for a pairwise sequence alignment is demonstrated below. If we have two sequences of DNA,

A T G G T C A G C
A C G G T C

Here the lengths of the sequences are 9 and 6 respectively. Therefore, the generated 2D array will have total of (10×7) = 70 nodes.

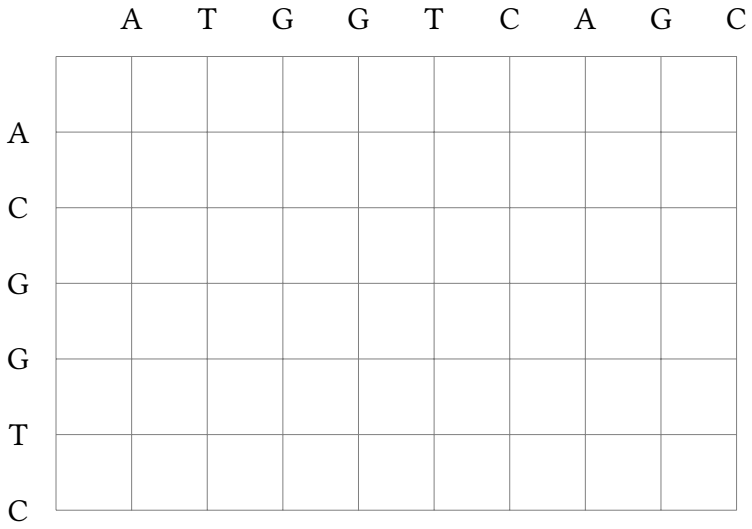


Figure 6.1: Edit graph

6.2 Generate Paths through the Edit Graph

After building a graph for the problem statement, next step is to generate paths through the graph. Here, every path corresponds to an alignment. But not all paths may correspond to the optimal alignment.
If the optimal path is figured out through 'Edit Distance' (Minimum number of operations required to change one sequence to another so that the sequences completely match each other), the paths with the lowest 'Edit Distance' (lowest number of mismatched characters) will correspond to the optimal path generated through the graph.
The transition (path) from one node to another should be either horizontal (right), vertical (below) or diagonal (right-below).

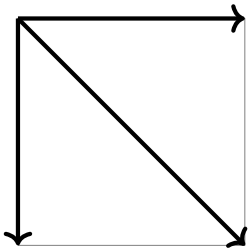


Figure 6.2: Possible transitions from a node

For the sequences in section 6.1, following figure demonstrates a generated optimal path though the graph which corresponds to the optimal alignment.

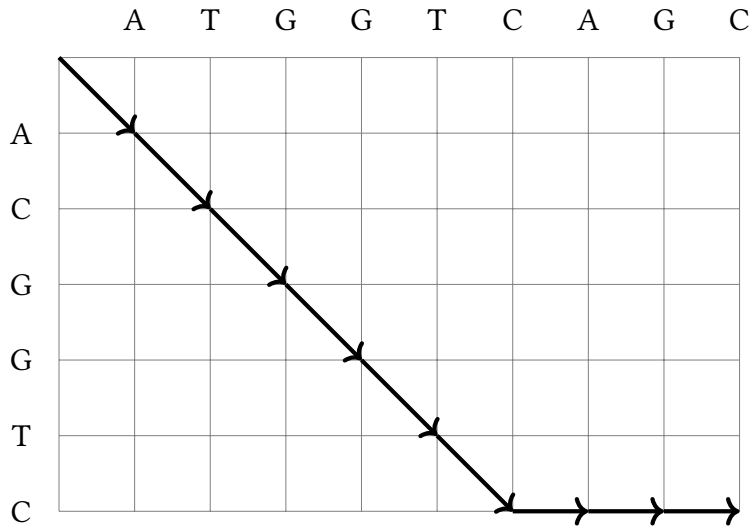


Figure 6.3: Optimal path generated through the edit graph

The marked path of the above graph corresponds to the following alignment,

A T G G T C A G C
A C G G T C _ _ _

The gaps correspond to "indel". Mismatch happens when characters of the same index do not match. The "edit operation" indicates three operations in case of mismatches or indels.

- Substitution is required for mismatches,
- Insertion is required in case of indels.
- Deletion is the process of removing a character.

The goal here is to minimize the differences between two sequences or in other words, generate a path through the graph with the minimum number of mismatches possible, considering all possible paths/combinations possible.

Chapter 7

Organization

7.1 Path Cost/Profit

In order to find out the optimal path, a proposition has been made on how the path cost/profit should be calculated. The particular method works by assigning cost/profit to each of the edges. In this thesis, instead of cost, profit shall be calculated based on the proposed oracle. The path with the highest profit shall correspond to the optimal path. A single block of the graph/grid shall act as a unit for profit calculation. For the following figure,

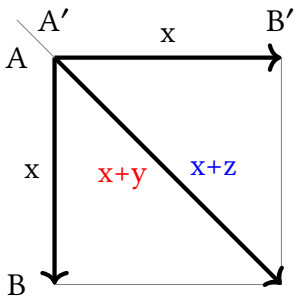


Figure 7.1: Possible path profits for single transition

For the figure above,

- x, y, z > 0
- z > y
- {A, A′, B, B′} ∈ {A, G, C, T}

Here,

1. If, transition from (A, A′) → (B, B′) and B = B′, then profit should be x+z.
2. If, transition from (A, A′) → (B, B′) and B ≠ B′, then profit should be x+y.
3. If, no transition from (A, A′) → (B, B′), then profit should be x.

Although, x, y and z can be any natural number, x, y, z ∈ ℕ with respect to the above conditions, throughout this thesis work, the value of x, y and z shall be 1, 1, 2 respectively to ensure simplicity in circuit building.

7.2 Quantum Implementation

To demonstrate the transition in a single block, a composite state of two qubits shall be used; the first qubit represents a horizontal transition and the second qubit represents a vertical transition. Here, the orientation can be vice versa as well.

Composite State	Transition	Profit	Direction
$\|00\rangle$	No Transition	0	•
$\|01\rangle$	Vertical (Lower)	x	↓
$\|10\rangle$	Horizontal (Right)	x	→
$\|11\rangle$	Diagonal (Lower Right)	x+y/x+z	↘

Table 7.1: Quantum representation of Transitions

7.2.1 Qubit Size

Number of Transitions

- If, there are a pair of sequences, whose length are m and n respectively then at most (m + n) transitions are required.
- If, m = n, then at least {(m + n) / 2} transitions are required.
- If, m ≠ n and m > n, then at least {n + (m - n)} transitions are required.
- If, m ≠ n and m < n, then at least {m + (n - m)} transitions are required.

If, t transitions are required to sort out the optimal path, t pairs of qubits or, simply $2t$ qubits are required to demonstrate every transitions.

Profit, Horizontal and Vertical Nodes

If there are a pair of sequences, whose length are m and n respectively then maximum profit should be,

- $3n$ or $3m$, if m = n.
- $3n + (m - n)$, if, m ≠ n and m > n.
- $3m + (n - m)$ otherwise.

The number of qubits required for profit calculation should be the minimum number of bits required for the binary representation of the maximum profit.
For the demonstration of horizontal and vertical shifts/transitions, number of qubits required shall be the minimum number of bits required to represent m and n.

The Letters in the Sequence

As, there can be four separate letters, A, T, G and C in the sequences, two qubits may suffice for the quantum representation. Although, there are no formal conventions, the table below can be followed to signify biological properties of DNA.

Composite State	Letter
$\|00\rangle$	A
$\|01\rangle$	C
$\|10\rangle$	G
$\|11\rangle$	T

Table 7.2: Quantum representation of Nucleotide Bases

Matching of Two Characters

It is required to match two characters in order to calculate profit. For instance, in Figure 6.1, it is needed to be verified if, B = B′ due to the transition from (A, A′) → (B, B′) and profit would be incremented based on the matching of the characters.
As, each of the characters (Nucleotide Bases) are represented with two qubits, two more qubits are required to act as control qubits along with another qubit which may act as the target qubit. If the target qubit is $|1\rangle$ then the characters match with each other.
If we take two 2-qubit sequences a and b, Toffoli gates that flips the target bit only when both the control states are either $|11\rangle$ or $|00\rangle$, are used in the following circuit to represent the matching process. In other words, the target qubit will be $|1\rangle$ if and only if both the control qubits are $|1\rangle$. Also, the control qubits would be $|1\rangle$ if $a = b$.

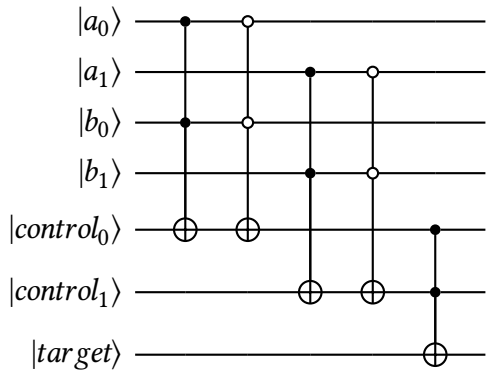


Figure 7.2: Quantum circuit to match two 2-bit numbers

7.3 Quantum Random Access Memory

In case of a classical computer, a RAM or Random Access Memory randomly addresses 2^n memory cells with n bits. A qRAM or quantum Random Access Memory can address the quantum superposition of 2^n memory cells with n qubits. The concept of qRAM has been proposed by Giovannetti, Lloyd and Maccone (2008) [33].
The proposed qRAM accesses memory addresses coherently using quantum superposition. To access a superposition of memory cells, the superposition of addresses must be inherited by an address register 'a' and through a superposition of data inherited by a data register 'd', qRAM passes the superposition of data to the quantum computer which needed to access the superposition of memory cells.
If we have two registers; an address register $|j\rangle$ and a data register $|D_j\rangle$,

$$\sum_j \psi|j\rangle_a \qquad \xrightarrow{qRAM} \qquad \sum_j \psi|j\rangle_a |D_j\rangle_d \tag{7.1}$$

Here, $\sum_j \psi|j\rangle_a$ corresponds to the superposition of addresses. The data D_j is stored in the jth address/location of the memory cell.

$$|j\rangle \rightarrow |D_j\rangle \tag{7.2}$$

To match sequences, two separate qRAM's are needed in order to generate specific characters/letters from the pair of DNA sequences and should correspond to a specific address/index of the address register.
For example considering the following pairs of addresses and data,

$$0 \rightarrow 1 \text{ (01 in binary)}$$
$$1 \rightarrow 3 \text{ (11 in binary)}$$

The corresponding qRAM is as follows,

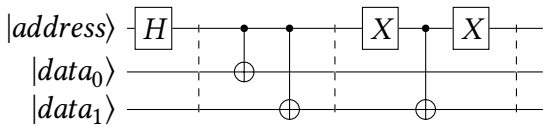


Figure 7.3: Sample qRAM

Chapter 8

Circuit and Algorithm

8.1 Path Generation and Index Calculation

By applying Hadamard gate to the pairs of transition qubits, we can generate all possible paths through the property of quantum superposition. The path profits are calculated simultaneously along the process.
Total number of steps required are equal to the number of total pair of transition qubits (One qubit corresponds to the horizontal shift and another corresponds to the vertical shift). Path generation for a single step is shown in figure 8.1.
Here, to match two characters (horizontal and vertical) in case of any lower diagonal transition, it is required to keep track of their particular indexes. We are using two counters which increments along with every corresponding transition.

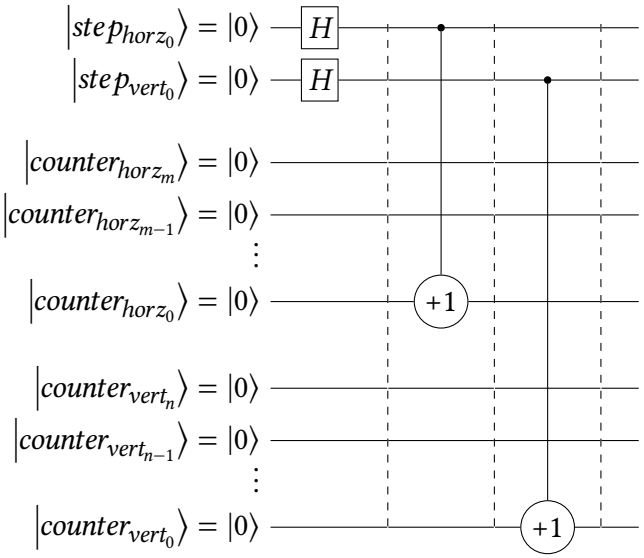


Figure 8.1: Path generation and counter incrementation for a single step

8.2 Quantum Adder Circuit

Although there are several methods and circuits for addition operations in quantum computers, we are referring to Draper (2000) [34].
Primary reason behind choosing this approach is for being convenient in terms of qubit

size reduction and deduction of the necessity of carry bits. This method can be implemented using the concept of Quantum Fourier Transform (QFT).

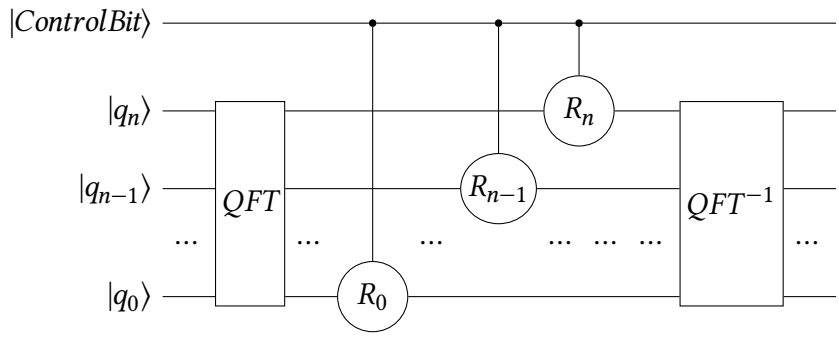


Figure 8.2: Quantum adder circuit

To add two numbers, the algorithm works as follows:

1. Apply QFT on any of the numbers.
2. Apply controlled phase gates (Transform Addition) on the transformed qubits. The rest of the qubits should act as control bits. Here, phase gates should operate as conditional rotation matrices,

$$R_k = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & e^{\frac{i\pi}{2^k}} \end{bmatrix} \tag{8.1}$$

3. Perform IQFT (Inverse Quantum Fourier Transform) to acquire the result of addition.

In this work, the maximum profit from a single step is equal to 1.

8.3 Circuit for matching characters

To match two separate characters, the proposed design requires four qubits as each character require two qubits. To get any character corresponding to a specific index, two qRAM's are applied; for horizontal and vertical sequences accordingly.
A sample demonstration is provided in figure 8.3. Both of the target bits would be $|1\rangle$ if the characters matches each other.

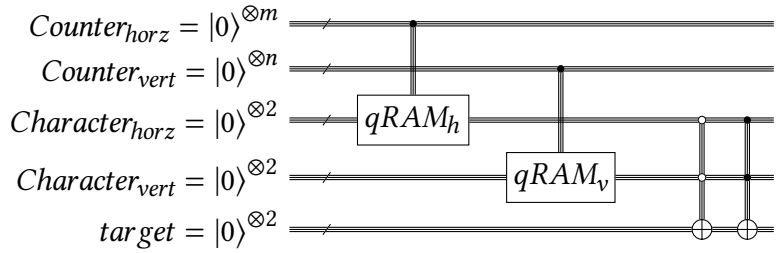


Figure 8.3: Circuit for matching two characters using qRAM

8.4 Profit Calculation

Completed circuit for calculating path along with the generated path is given below. This approach ensures the traversal/visit to every possible nodes in contrast to probabilistic/heuristic approaches. Also, total number of steps are also not increased and therefore, can figure out the optimal alignment using deterministic approach with nearly full precision.
As mentioned previously, if there are *t* number of transitions/steps, total *2t* registers are required for those steps. For the counters, number of qubits will be equal to the number of bits required for the binary representation of the length of the horizontal and vertical sequence.

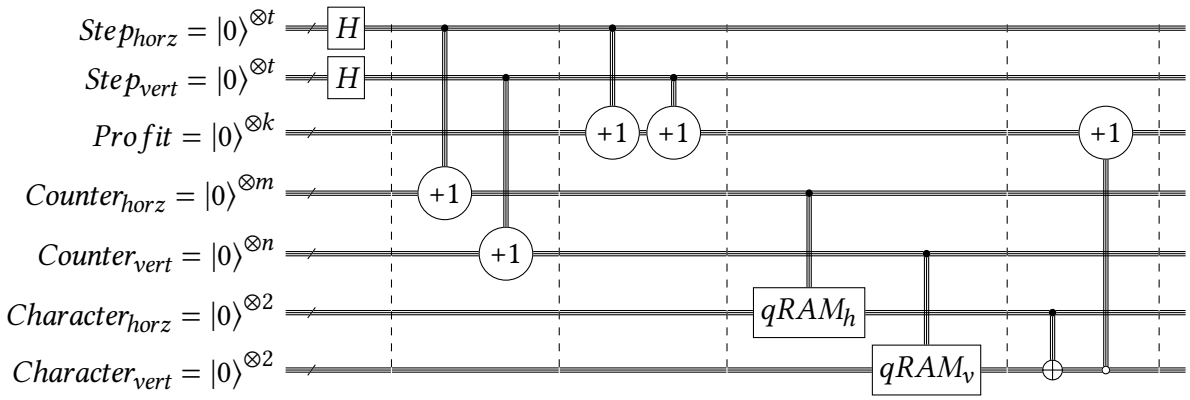


Figure 8.4: Quantum circuit to generate path and calculate profit

8.5 Finding the Path with Maximum Profit

To find out the optimal path, this thesis work is focusing on the approach taken by Ahuja and Kapoor (1999) [19]. The proposed method uses Grover's search algorithm to find out the maximum element from an unsorted array.
The algorithm for finding out the maximum profit:

1. Start with any initial guess of an index *a* from an array *D* of length N, such that a $\in$ (0, ..., N - 1).

2. Repeat a loop for O($\sqrt{N}$) times:

   - Take a initialized state using n-bit Hadamard transfomation;
     $|\psi\rangle = \sum_i \frac{1}{\sqrt{N}} |i\rangle |a\rangle$. *i* is the index of the maximum element.

- Find marked states using Grover's algorithm such that the following oracle's are satisfied.

$$f_i(j) = 1, \;\; if \;\; D[j] > D[i] \;\; and \;\; f_a(x) = 1 \tag{8.2}$$

- Make measurements. Replace a with the result of the measurements to make a new guess.

3. Return the index of the maximum element.

8.6 Complete Circuit

The complete circuit would be the merge between modified circuit of Grover's algorithm for finding out the maximum element from an unsorted array and the circuit for calculating profit by generating paths through the edit graph. Full circuit demonstration is provided in figure 8.5.

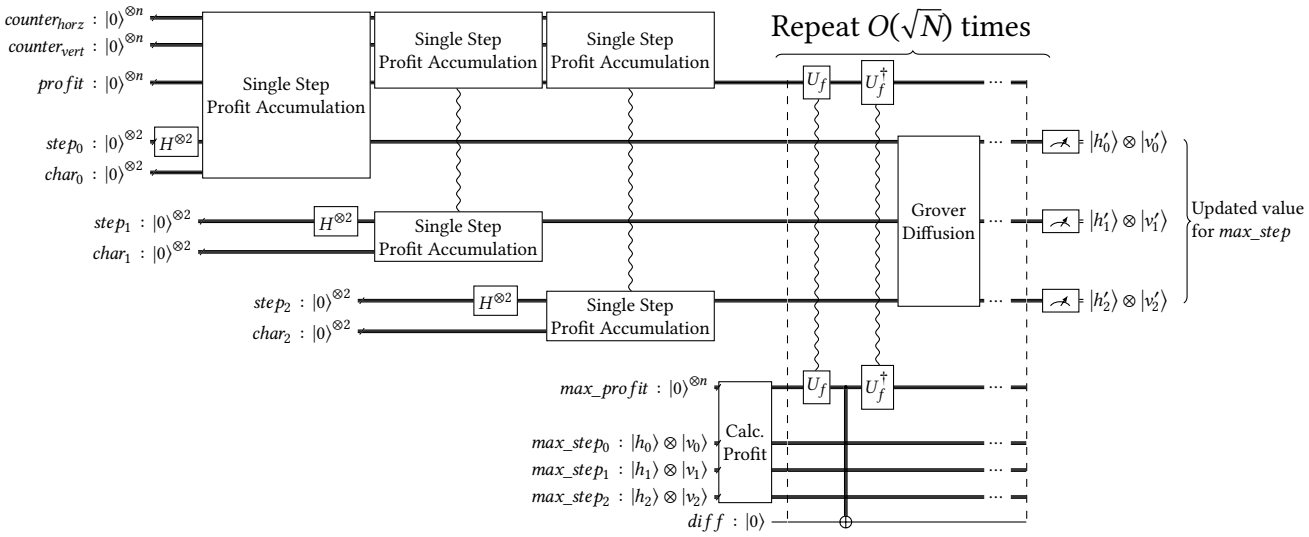


Figure 8.5: Quantum Circuit for Pairwise Sequence Alignment

8.7 Complexity

For our scope of the work in this paper, we are not considering the space complexity due to the implementation of qRAM in the circuit.
In the first section of generating the paths and calculating profit, N (or, M, if N is the length of the first sequence and M is the length of the second sequence.) steps would be required on average. To find out the optimal path, the searching process based on Grover's algorithm requires $\sqrt{N}$ steps [7]. Let, N be the average length of the sequences. Total number of steps will be the total time required by both the processes. Therefore, time complexity of the algorithm is $O(N) + O(\sqrt{N})$. After eliminating the non-dominant term it turns to be $O(N)$.

Chapter 9

Conclusion

DNA Sequence alignment is one of the topics of extensive research in computational biology and bio-informatics. In this work, we proposed a quantum algorithm to find out the optimal DNA sequences by aligning these sequences. In order to do so, we had to implement a deterministic method based on a graph traversing problem and Grover's search algorithms. Moreover, analyzing our proposed method and calculate the time and memory requirements has been a major task. Although, to find out the possible advantages of quantum computation in bio-informatics and computational biology, we are still required to compare the existing classical algorithms with the proposed quantum counterpart with full precision. In summary, our primary focus has been to create a graph to align the DNA sequences, generate random paths through it based on the path transition profits, implement a quantum algorithm and sort out the optimal DNA sequences.

Bibliography

[1] S. Aaronson, *Quantum Computing Since Democritus*. Cambridge University Press, 2013.

[2] M. Gollery, "Bioinformatics: Sequence and genome analysis," *Clinical Chemistry*, vol. 51, no. 11, pp. 2219–2220, 2005.

[3] J. D. Hidary, *Quantum Computing: An Applied Approach*. Springer, 2021, vol. 1.

[4] N. S. Yanofsky and M. A. Mannucci, *Quantum Computing for Computer Scientists*. Cambridge University Press, 2008.

[5] M. A. Nielsen and I. Chuang, *Quantum computation and quantum information*, 2002.

[6] S. B. Needleman and C. D. Wunsch, "A general method applicable to the search for similarities in the amino acid sequence of two proteins," *Journal of Molecular Biology*, vol. 48, no. 3, pp. 443–453, 1970.

[7] L. K. Grover, "A fast quantum mechanical algorithm for database search," in *Proceedings of the twenty-eighth annual ACM symposium on Theory of computing*, 1996, pp. 212–219.

[8] N. S. Manton, "The principle of least action in dynamics," *Cambridge University Press*, 2013.

[9] J. F. L.N. Hand, "Analytical mechanics," *Cambridge University Press*, 2008.

[10] R. D. Klauber, "Path integrals in quantum theories: A pedagogic first step," 2013.

[11] B. Alberts, A. Johnson, J. Lewis, M. Raff, K. Roberts, and P. Walter, *Molecular Biology of The Cell*. New York: Garland Science USA, 2002, vol. 4.

[12] P. J. Russell and K. Gordey, *IGenetics*. Benjamin Cummings San Francisco, 2002.

[13] D. Mount, *Bioinformatics: Sequence and genome analysis*, 2004.

[14] X. Huang and K.-M. Chao, "A generalized global alignment algorithm," *Bioinformatics*, vol. 19, no. 2, pp. 228–233, 2003.

[15] V. O. Polyanovsky, M. A. Roytberg, and V. G. Tumanyan, "Comparative analysis of the quality of a global algorithm and a local algorithm for alignment of two sequences," *Algorithms for Molecular Biology*, vol. 6, no. 1, pp. 1–12, 2011.

[16] M. J. Zvelebil and J. O. Baum, *Understanding Bioinformatics*. Garland Science, 2007.

[17] L. Wang and T. Jiang, "On the complexity of multiple sequence alignment," *Journal of computational biology*, vol. 1, no. 4, pp. 337–348, 1994.

[18] J. Keith and D. P. Kroese, "Sequence alignment by rare event simulation," *Proceedings of the Winter Simulation Conference*, vol. 1, pp. 320–327, 2002. DOI: 10.1109/WSC.2002.1172901.

[19] A. Ahuja and S. Kapoor, "A quantum algorithm for finding the maximum," *arXiv preprint quant-ph/9911082*, 1999.

[20] T. F. Smith and M. S. Waterman, "Identification of common molecular subsequences," *Journal of molecular biology*, vol. 147, no. 1, pp. 195–197, 1981.

[21] C. E. Lawrence, S. F. Altschul, M. S. Boguski, J. S. Liu, A. F. Neuwald, and J. C. Wootton, "Detecting subtle sequence signals: A gibbs sampling strategy for multiple alignment," *science*, vol. 262, no. 5131, pp. 208–214, 1993.

[22] P. A. Pevzner, "Multiple alignment, communication cost, and graph matching," *SIAM Journal on Applied Mathematics*, vol. 52, no. 6, pp. 1763–1779, 1992.

[23] S. E. Venegas-Andraca, "Quantum walks: A comprehensive review," *Quantum Information Processing*, vol. 11, no. 5, pp. 1015–1106, 2012.

[24] N. B. Lovett, S. Cooper, M. Everitt, M. Trevers, and V. Kendon, "Universal quantum computation using the discrete-time quantum walk," *Physical Review A*, vol. 81, no. 4, p. 042 330, 2010.

[25] D. Aharonov, A. Ambainis, J. Kempe, and U. Vazirani, "Quantum walks on graphs," in *Proceedings of the thirty-third annual ACM symposium on Theory of computing*, 2001, pp. 50–59.

[26] F. Sánchez, E. Salami, A. Ramirez, and M. Valero, "Performance analysis of sequence alignment applications," in *2006 IEEE International Symposium on Workload Characterization*, IEEE, 2006, pp. 51–60.

[27] A. Tulsi, "Faster quantum-walk algorithm for the two-dimensional spatial search," *Physical Review A*, vol. 78, no. 1, p. 012 310, 2008.

[28] C. Dürr, M. Heiligman, P. HOyer, and M. Mhalla, "Quantum query complexity of some graph problems," *SIAM Journal on Computing*, vol. 35, no. 6, pp. 1310–1328, 2006.

[29] M. Santha, "Quantum walk based search algorithms," pp. 31–46, 2008.

[30] A. Ambainis, "Quantum walks and their algorithmic applications," *International Journal of Quantum Information*, vol. 1, no. 04, pp. 507–518, 2003.

[31] A. Sarkar, Z. Al-Ars, C. G. Almudever, and K. Bertels, "An algorithm for dna read alignment on quantum accelerators," *arXiv preprint arXiv:1909.05563*, 2019.

[32] K. Prousalis and N. Konofaos, "A quantum pattern recognition method for improving pairwise sequence alignment," *Scientific reports*, vol. 9, no. 1, pp. 1–11, 2019.

[33] V. Giovannetti, S. Lloyd, and L. Maccone, "Quantum random access memory," *Phys. Rev. Lett.*, vol. 100, p. 160 501, 16 Apr. 2008. DOI: 10.1103/PhysRevLett.100.160501. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevLett.100.160501.

[34] T. G. Draper, "Addition on a quantum computer," *arXiv preprint quant-ph/0008033*, 2000.