Grover's Algorithm and the Molecular Distance Geometry Problem

Avi Swartz

ES 170 - April 30, 2020

SUBMITTED FOR PUBLICATION TO: ?CBRN 2005 Conference, AUGUST 17, 2005

Grover's Algorithm applied to the Molecular Distance Geometry Problem

Carlile Lavor, Leo Liberti and Nelson Maculan

***Abstract*—Grover's algorithm is a quantum algorithm for searching in unstructured databases. Due to the properties of quantum mechanics, it provides a quadratic speedup over its classical counterparts. We present a new application of the algorithm to the Molecular Distance Geometry Problem. This problem is related to the determination of the tridimensional structure of a molecule based on the knowledge of some of the distances between pairs of atoms. This problem is NP-hard unless all possible inter-atomic distances are known.**

***Index Terms*—Quantum Computation, Grover's algorithm, Molecular Distance Geometry Problem.**

II. The Molecular Distance Geometry Problem

Formally, the MDGP can be defined as the problem of finding Cartesian coordinates $x_1, \ldots, x_n \in \mathbb{R}^3$ of the atoms of a molecule such that for all $(i,j) \in S$,

$$||x_i - x_j|| = d_{ij},$$

where S is the set of pairs of atoms (i,j) whose Euclidean distances d_{ij} are known. If all distances are given, the problem can be solved in linear time [5]. Otherwise, the problem is NP-hard [16].

This algorithm was first published in this paper but never implemented or explored.

Grover's Algorithm

- Imagine you have some circuit that takes n bits and outputs 1 bit.
- This circuit implements some function $f : \{0,1\}^n \to \{0,1\}$

Grover's Algorithm

- Imagine you have some circuit that takes n bits and outputs 1 bit.
- This circuit implements some function $f : \{0,1\}^n \to \{0,1\}$
- Suppose you want to find strings x such that $f(x) = 1$

Grover's Algorithm

- Imagine you have some circuit that takes n bits and outputs 1 bit.
- This circuit implements some function $f : \{0,1\}^n \to \{0,1\}$
- Suppose you want to find strings x such that $f(x) = 1$
  - With a classical computer, you need to try all 2^n combinations

Grover's Algorithm

- Imagine you have some circuit that takes n bits and outputs 1 bit.
- This circuit implements some function $f : \{0,1\}^n \to \{0,1\}$
- Suppose you want to find strings x such that $f(x) = 1$
  - With a classical computer, you need to try all 2^n combinations
  - With a quantum computer, you only need to try $\frac{\pi}{4}2^{n/2}$ combinations
    - This is 10^{15} times faster for $n = 100$

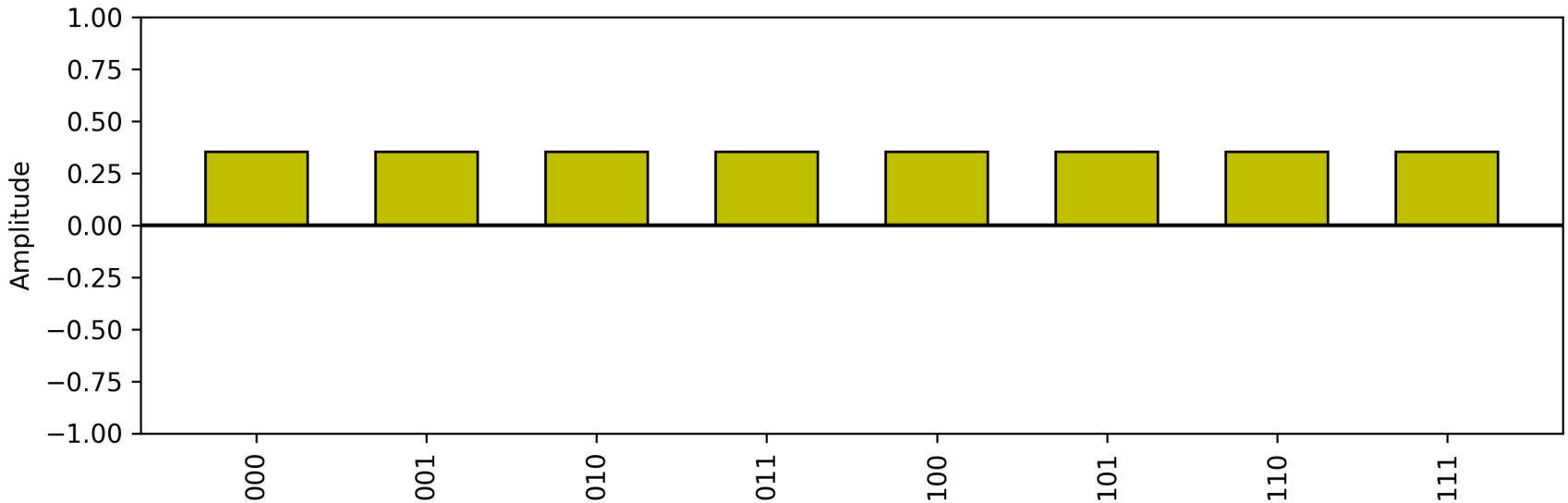
Grover's Algorithm

Implementation:

Grover's Algorithm

Implementation:

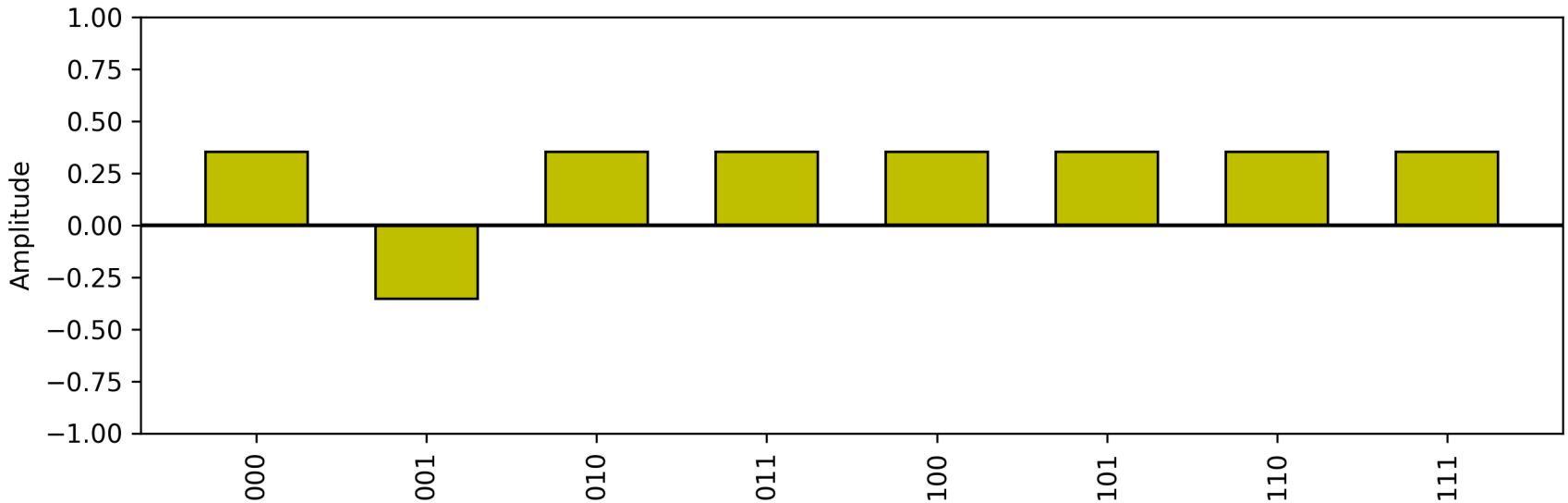
1. Put n qubits into an equal superposition



Grover's Algorithm

Implementation:

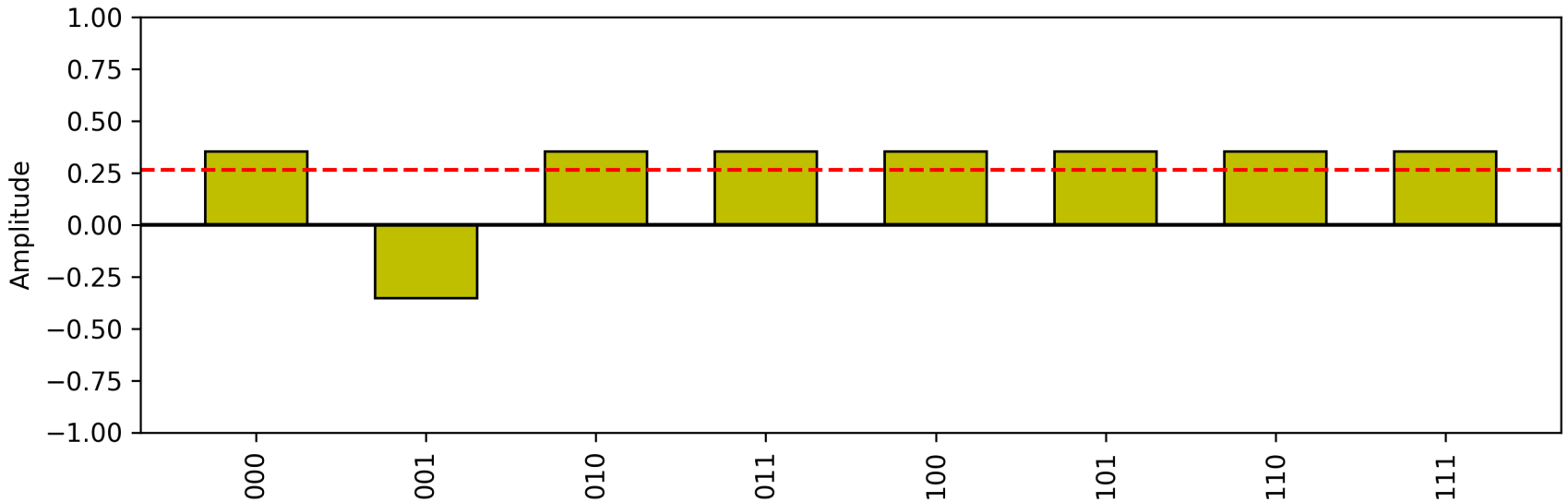
1. Put n qubits into an equal superposition
2. Use a quantum version of the f circuit to negate the amplitude of all states x where $f(x) = 1$



Grover's Algorithm

Implementation:

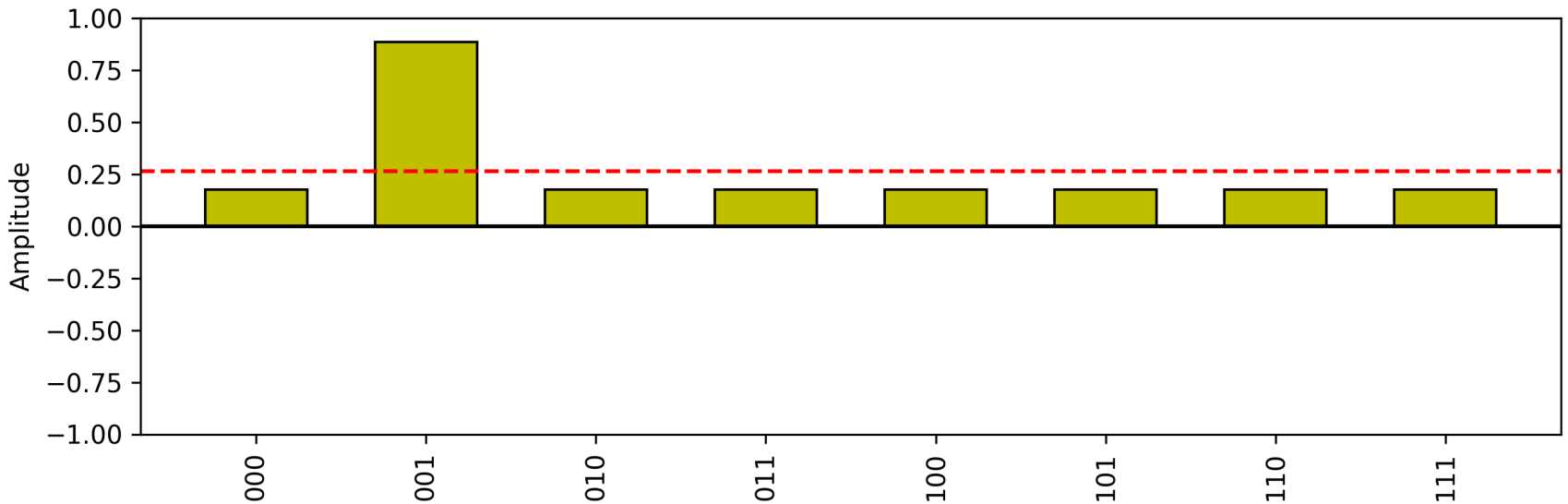
1. Put n qubits into an equal superposition
2. Use a quantum version of the f circuit to negate the amplitude of all states x where $f(x) = 1$
3. Take the mean amplitude



Grover's Algorithm

Implementation:

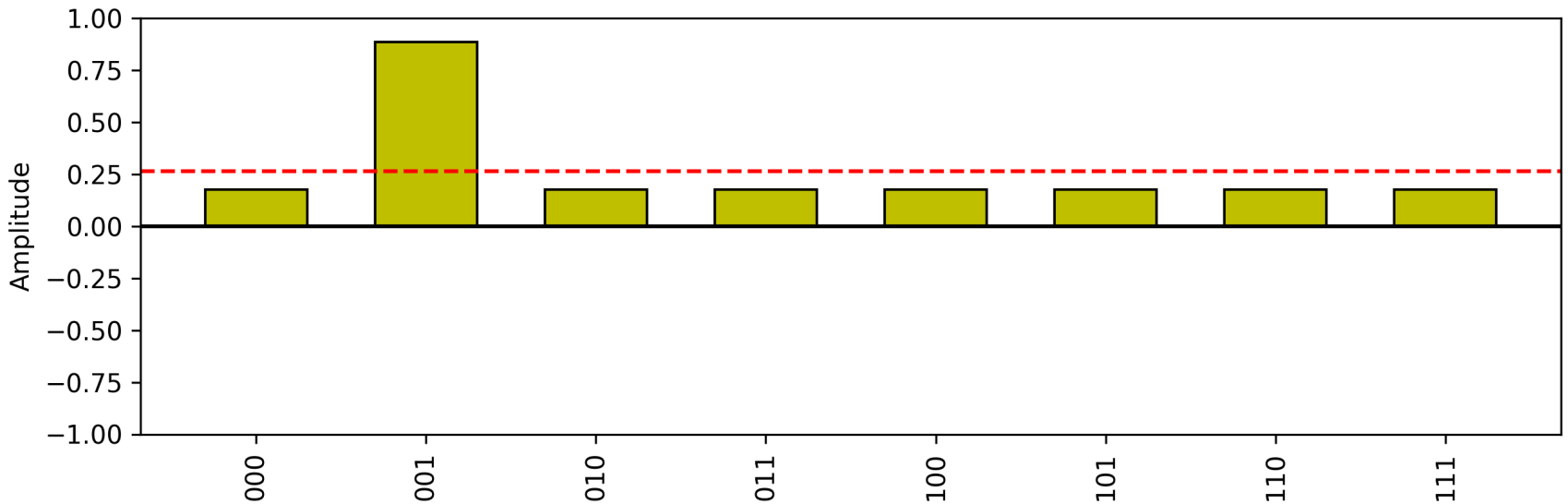
1. Put n qubits into an equal superposition
2. Use a quantum version of the f circuit to negate the amplitude of all states x where $f(x) = 1$
3. Take the mean amplitude
4. Invert all amplitudes around the mean



Grover's Algorithm

Implementation:

1. Put n qubits into an equal superposition
2. Use a quantum version of the f circuit to negate the amplitude of all states x where $f(x) = 1$
3. Take the mean amplitude
4. Invert all amplitudes around the mean
5. Repeat steps 2-4 exactly $\frac{\pi}{4}2^{n/2}$ times.



Molecular Distance Geometry Problem (MDGP)

Molecular Distance Geometry Problem (MDGP)

- You have a bunch of atoms

Molecular Distance Geometry Problem (MDGP)

- You have a bunch of atoms
- You know the distances between some of the pairs of atoms (NMR does this very nicely)

Molecular Distance Geometry Problem (MDGP)

- You have a bunch of atoms
- You know the distances between some of the pairs of atoms (NMR does this very nicely)
- You want to know the 3D coordinates of all the atoms so that all the distances are satisfied

Molecular Distance Geometry Problem (MDGP)

- You have a bunch of atoms
- You know the distances between some of the pairs of atoms (NMR does this very nicely)
- You want to know the 3D coordinates of all the atoms so that all the distances are satisfied
- Unfortunately, if you don't have the distances between all the pairs of atoms, this is NP-Hard

  (i.e. it's really hard, and if you solved this, you would also solve a lot of other really hard problems and it would be a really big deal)

Grover's on the MDGP

If you know the following about your molecule:

Grover's on the MDGP

If you know the following about your molecule:

- All the atoms form a chain from beginning to end

Grover's on the MDGP

If you know the following about your molecule:

- All the atoms form a chain from beginning to end
- You know the distance between each pair of consecutive atoms on the chain

Grover's on the MDGP

If you know the following about your molecule:

- All the atoms form a chain from beginning to end
- You know the distance between each pair of consecutive atoms on the chain
- You know the angle between every three atoms on the chain

Grover's on the MDGP

If you know the following about your molecule:

- All the atoms form a chain from beginning to end
- You know the distance between each pair of consecutive atoms on the chain
- You know the angle between every three atoms on the chain
- You know the distance between every pair of atoms 3 bonds apart on the chain

then you can

Grover's on the MDGP

If you know the following about your molecule:

- All the atoms form a chain from beginning to end
- You know the distance between each pair of consecutive atoms on the chain
- You know the angle between every three atoms on the chain
- You know the distance between every pair of atoms 3 bonds apart on the chain

then you can

take the first atom,

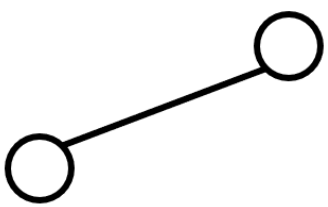
Grover's on the MDGP

If you know the following about your molecule:

- All the atoms form a chain from beginning to end
- You know the distance between each pair of consecutive atoms on the chain
- You know the angle between every three atoms on the chain
- You know the distance between every pair of atoms 3 bonds apart on the chain

then you can

place the second atom the known distance away



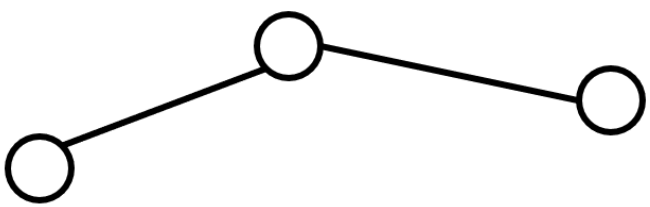
Grover's on the MDGP

If you know the following about your molecule:

- All the atoms form a chain from beginning to end
- You know the distance between each pair of consecutive atoms on the chain
- You know the angle between every three atoms on the chain
- You know the distance between every pair of atoms 3 bonds apart on the chain

then you can

place the third atom the known distance and angle away



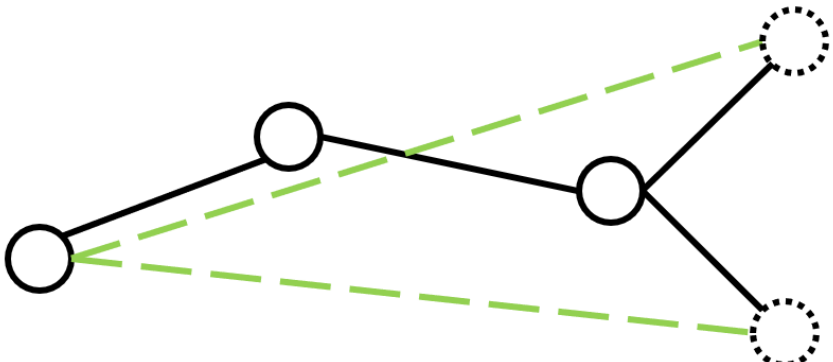
Grover's on the MDGP

If you know the following about your molecule:

- All the atoms form a chain from beginning to end
- You know the distance between each pair of consecutive atoms on the chain
- You know the angle between every three atoms on the chain
- You know the distance between every pair of atoms 3 bonds apart on the chain

then you can

have two places to put the fourth atom, since you know how far it is from the third atom, the angle it makes with the third atom, and how far it is from the first atom



Grover's on the MDGP

Grover's on the MDGP

- You have two possibilities for the 4^{th} atom, and similarly then two possibilities for every atom after that.

Grover's on the MDGP

- You have two possibilities for the 4^{th} atom, and similarly then two possibilities for every atom after that.
- Represent the molecule with a binary string.
  - 1011 is the molecule where the 4th atom goes up, 5th goes down, 6th and 7th go up

Grover's on the MDGP

- You have two possibilities for the 4^{th} atom, and similarly then two possibilities for every atom after that.
- Represent the molecule with a binary string.
  - 1011 is the molecule where the 4th atom goes up, 5th goes down, 6th and 7th go up
- Let f be the function that takes a string and determines if the corresponding molecule satisfies the known distances

Grover's on the MDGP

- You have two possibilities for the 4^{th} atom, and similarly then two possibilities for every atom after that.
- Represent the molecule with a binary string.
  - 1011 is the molecule where the 4th atom goes up, 5th goes down, 6th and 7th go up
- Let f be the function that takes a string and determines if the corresponding molecule satisfies the known distances
- Grover's Algorithm!

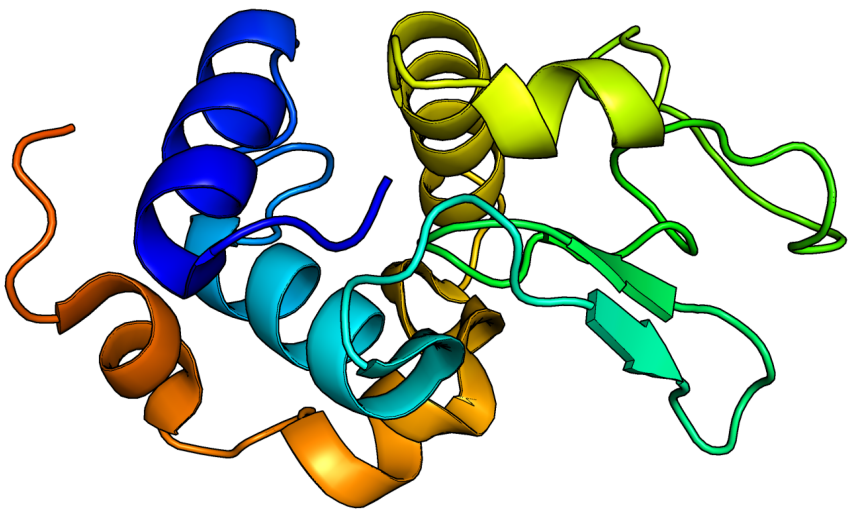
Grover's on the MDGP

Is the chain assumption realistic?

Grover's on the MDGP

Is the chain assumption realistic?

Proteins!



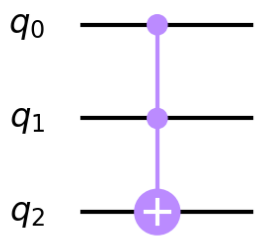
PDB 1DPX

Classical Gates

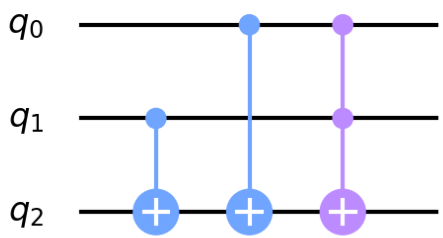
How do we implement the oracle?

Classical Gates

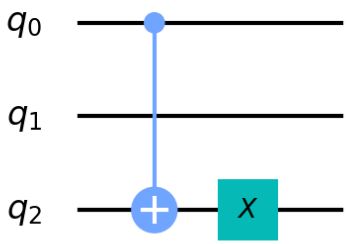
How do we implement the oracle?



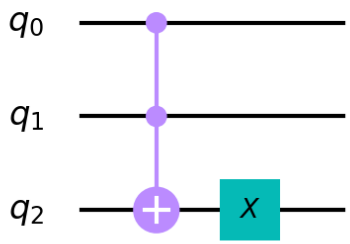
$$|xy0\rangle \rightarrow |xy(x \texttt{ and } y)\rangle$$



$$|xy0\rangle \rightarrow |xy(x \texttt{ or } y)\rangle$$



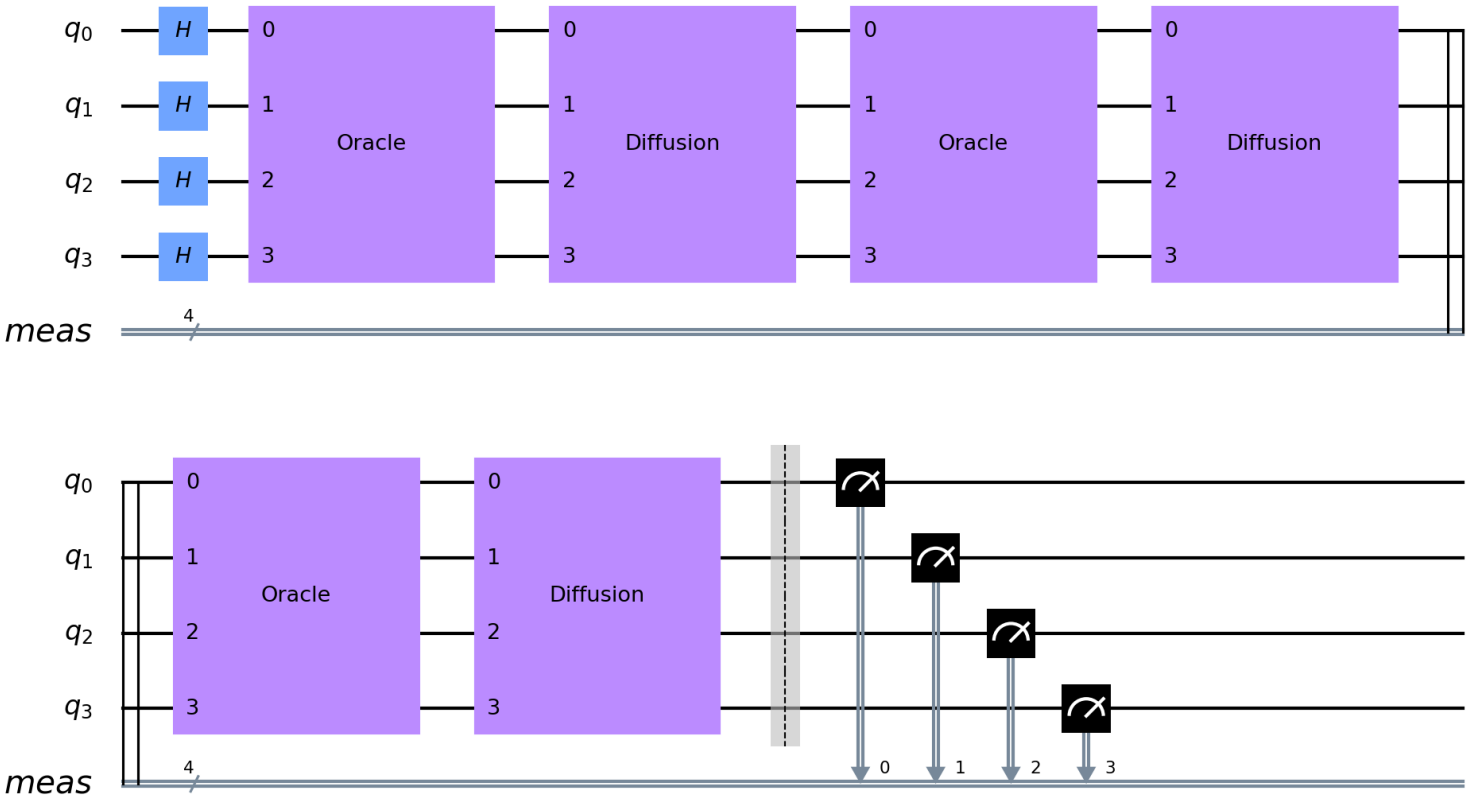
$$|xy0\rangle \rightarrow |xy(\texttt{not } x)\rangle$$



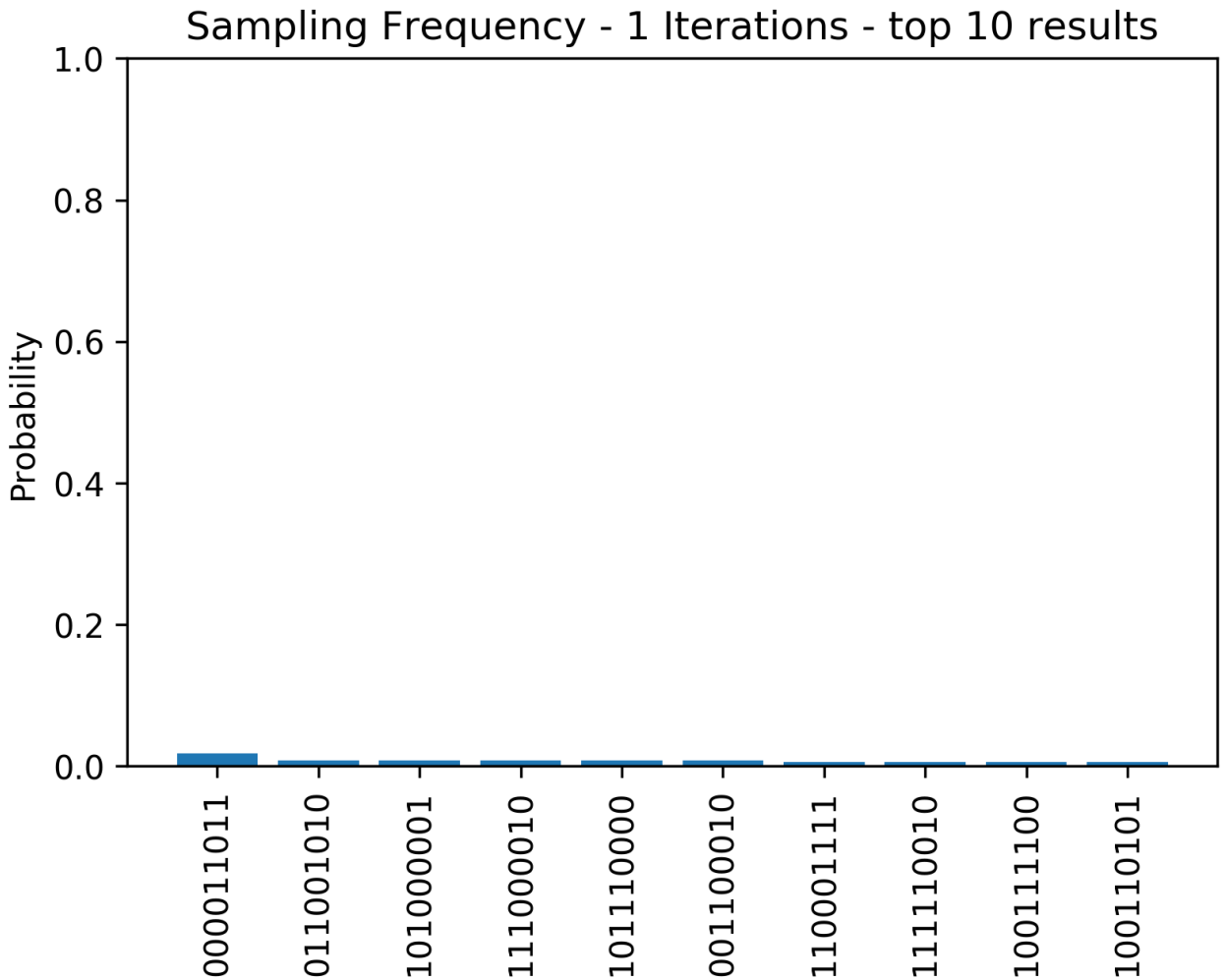
$$|xy0\rangle \rightarrow |xy(x \texttt{ nand } y)\rangle$$

Any classical computation can be easily performed on a quantum computer. Superpositions are maintained as well!

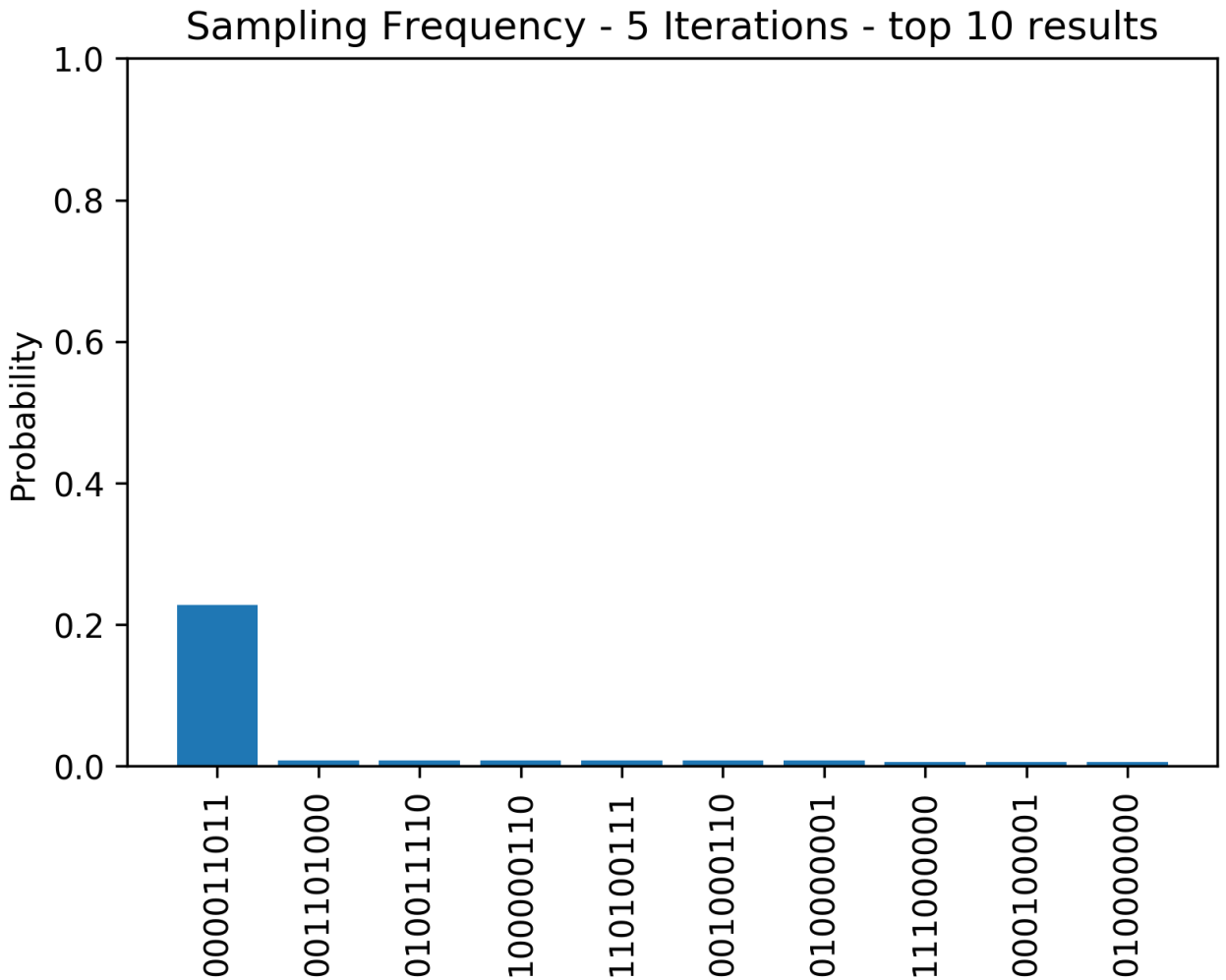
Implementation



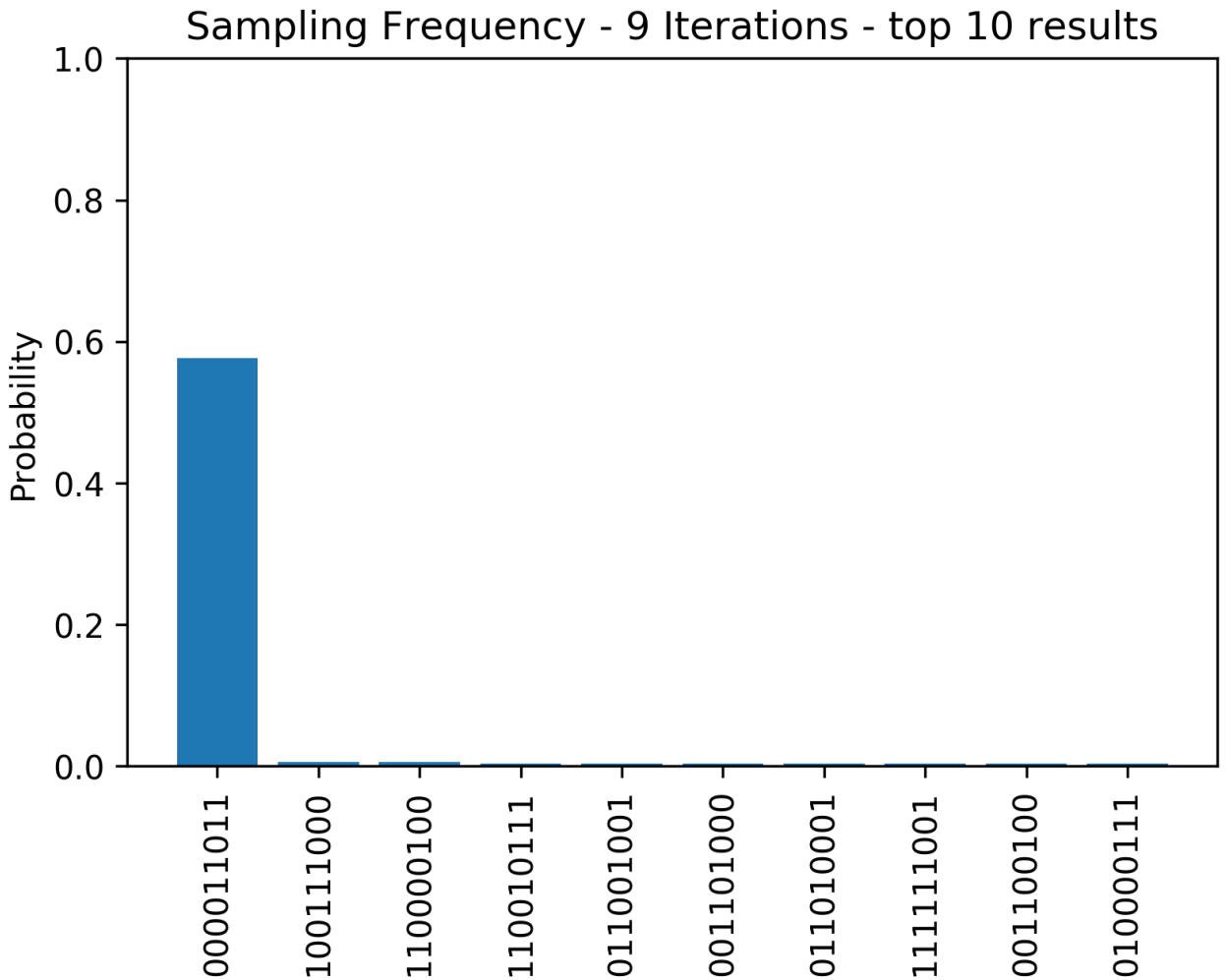
Sampling Results



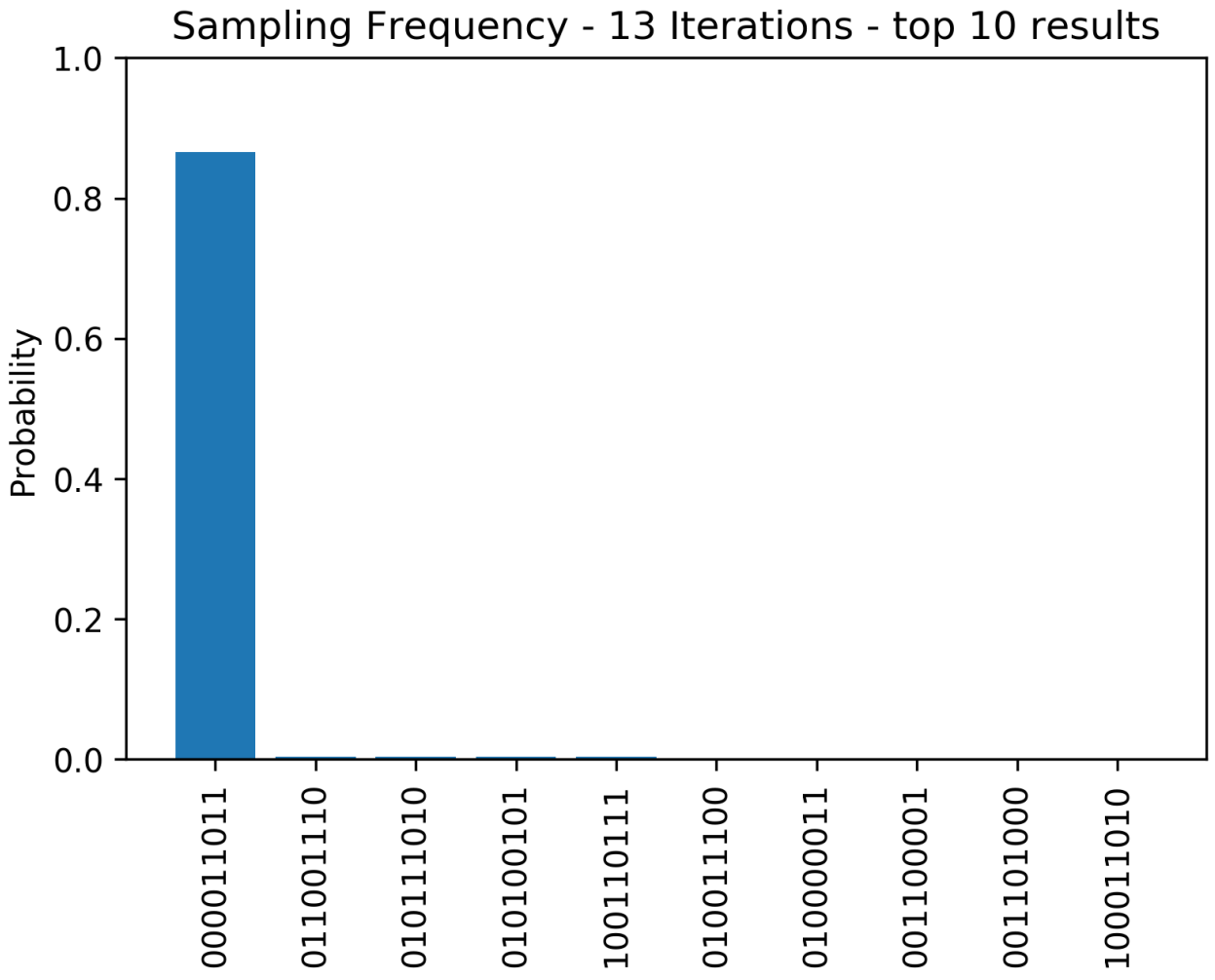
Sampling Results



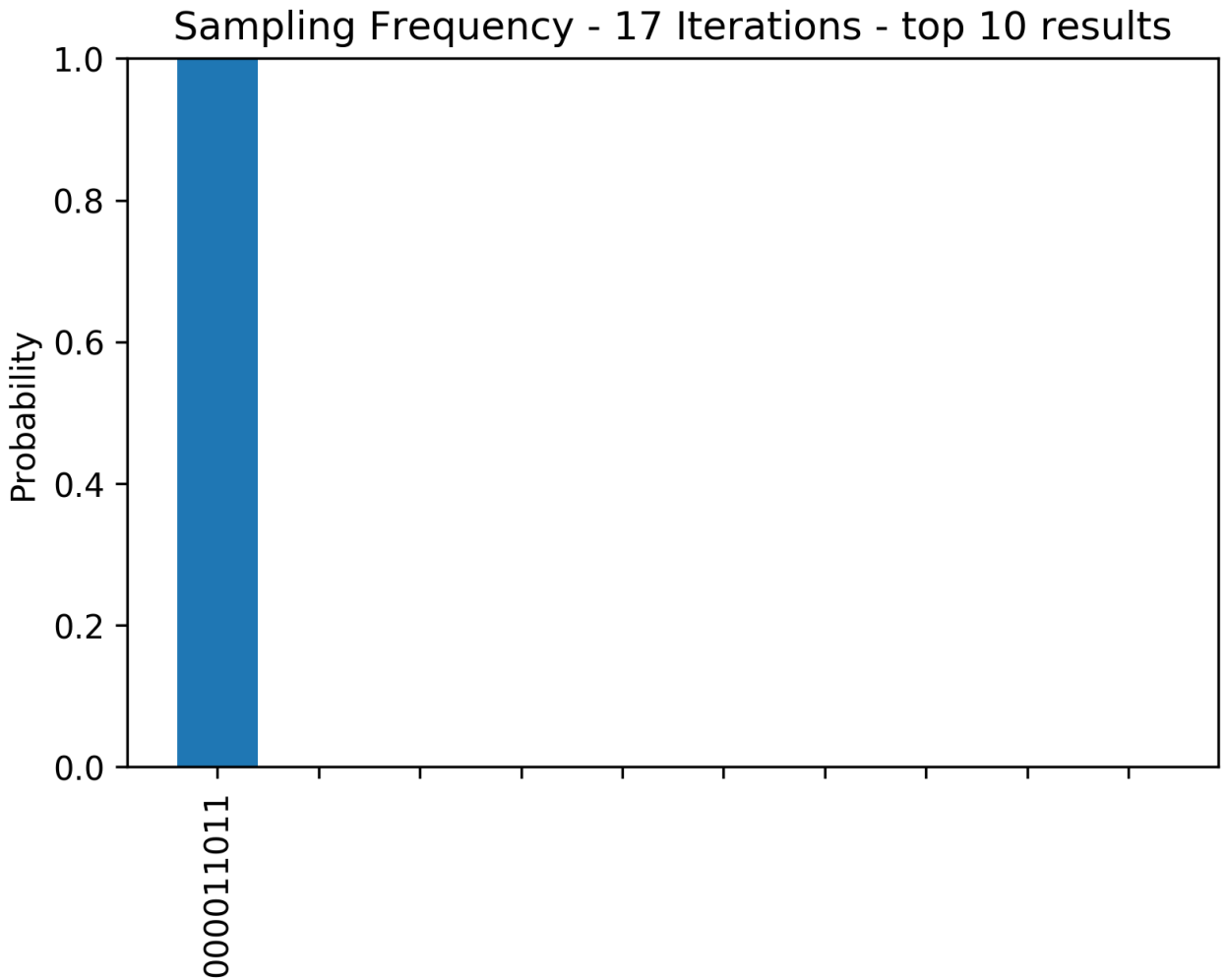
Sampling Results



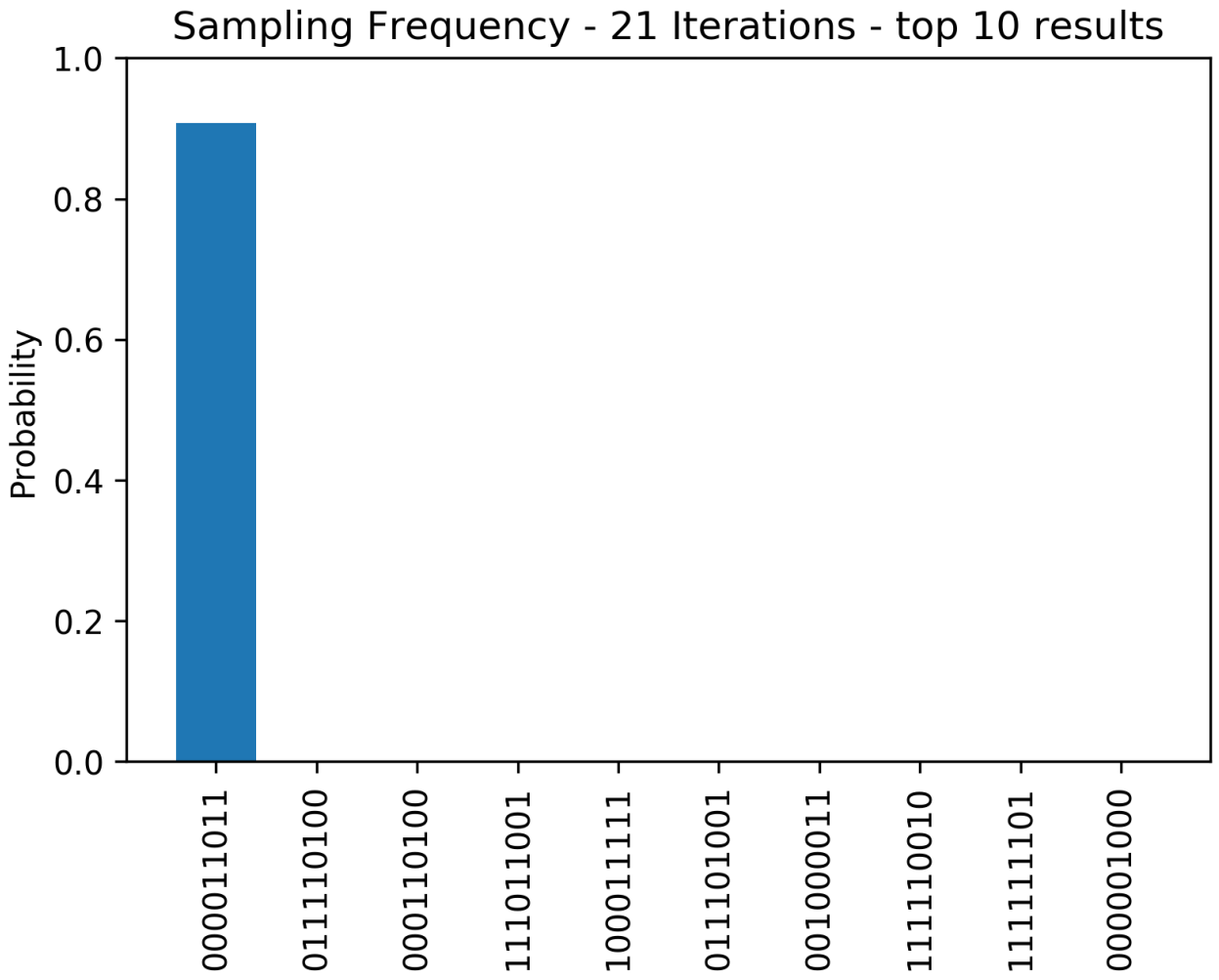
Sampling Results



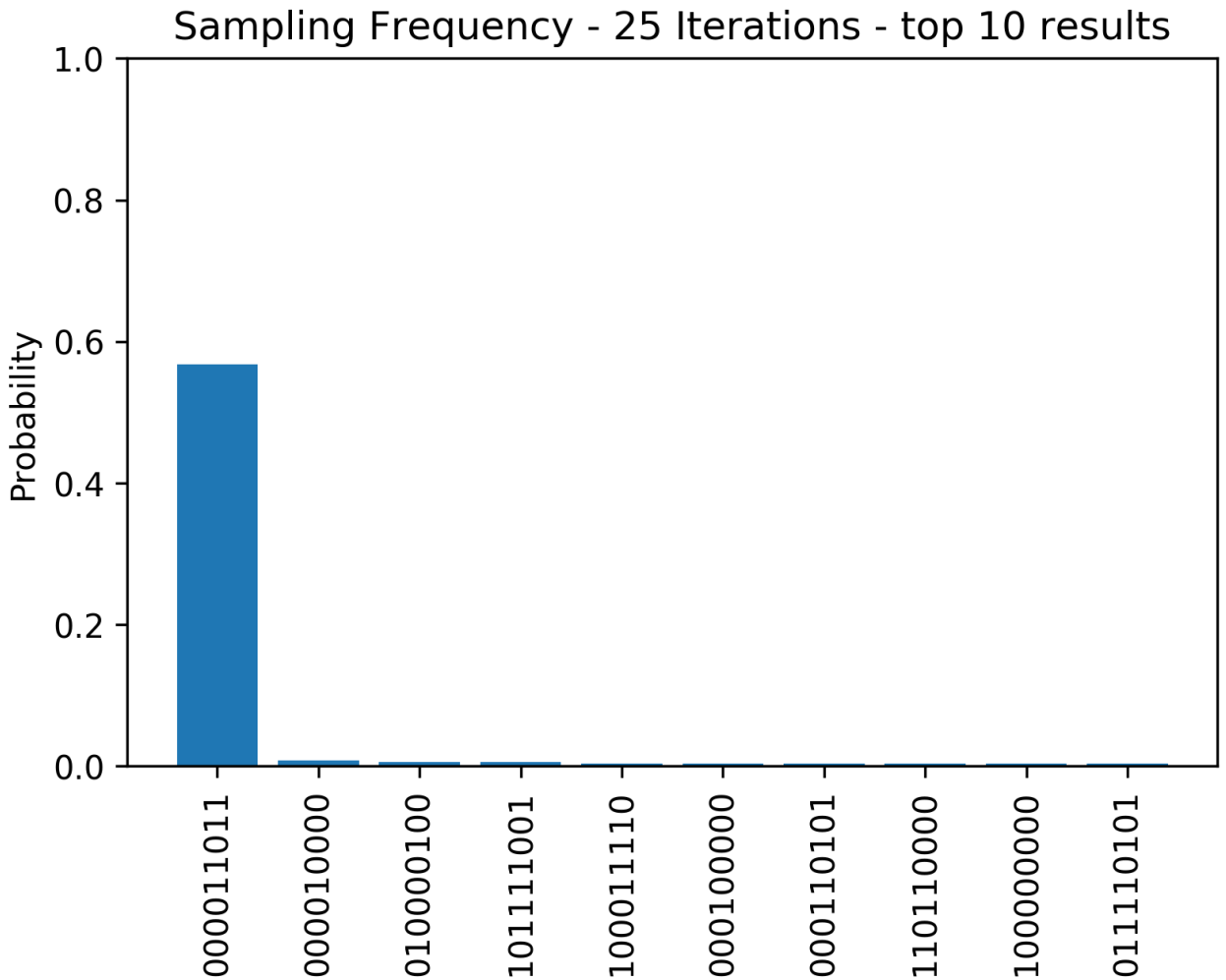
Sampling Results



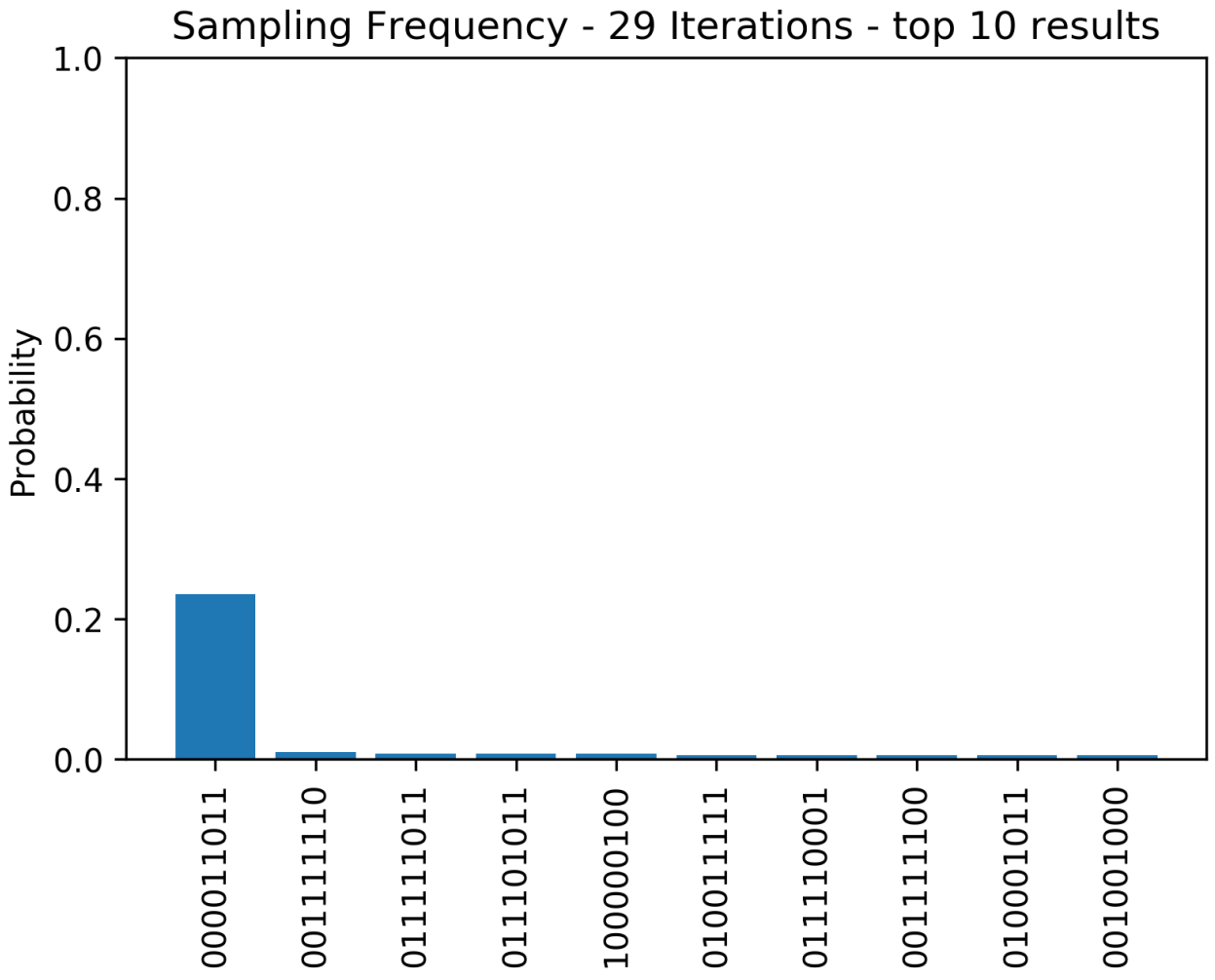
Sampling Results



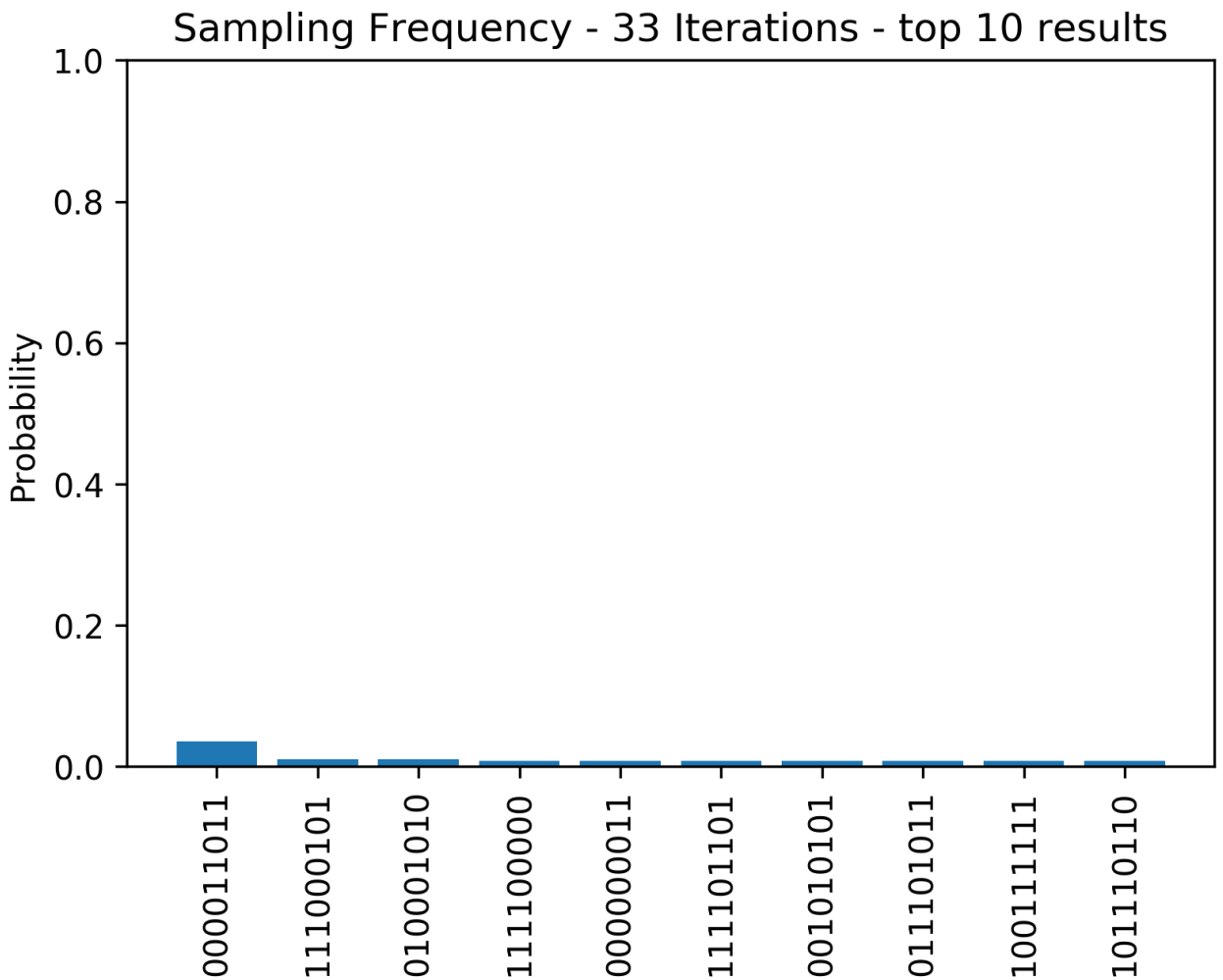
Sampling Results



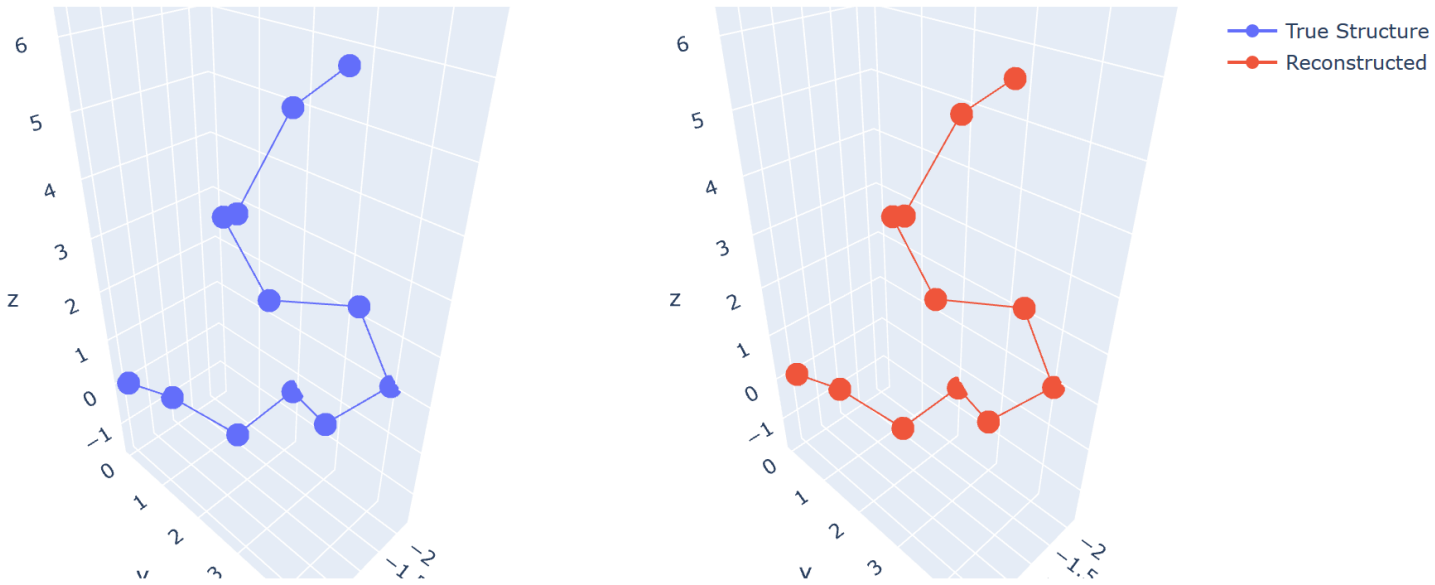
Sampling Results



Sampling Results

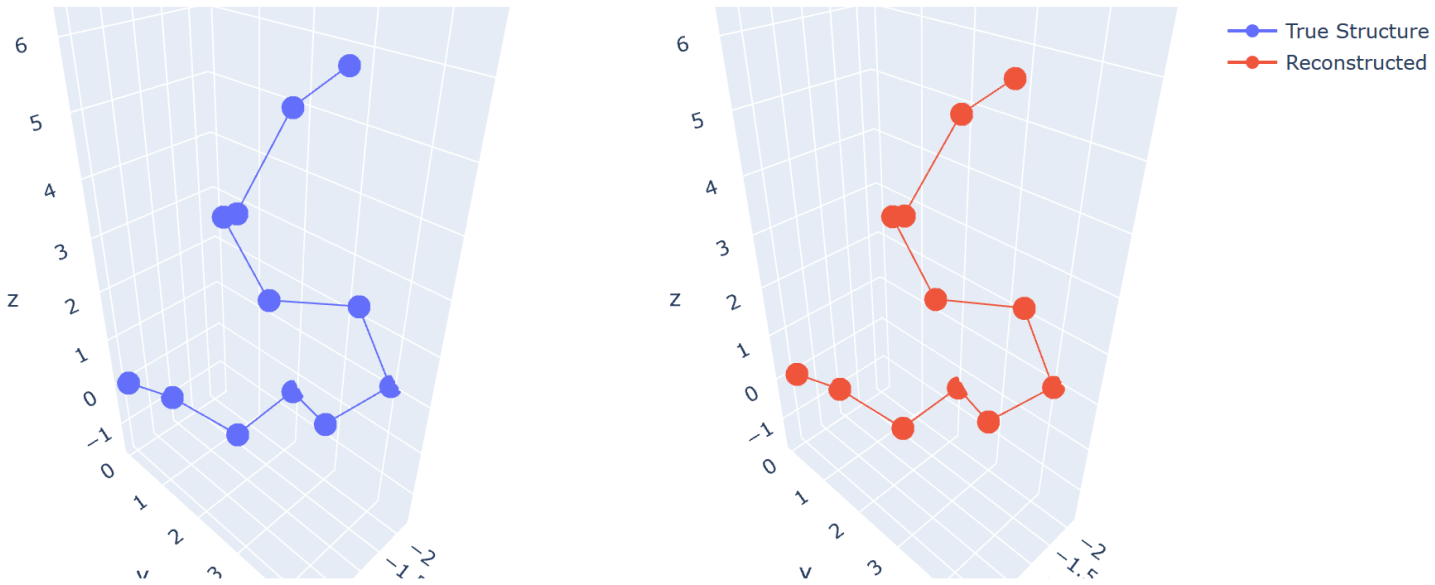


Small Reconstruction



The algorithm works great when you have time to implement $\frac{\pi}{4}2^{n/2}$ rounds of Grover's algorithm.

Small Reconstruction



The algorithm works great when you have time to implement $\frac{\pi}{4}2^{n/2}$ rounds of Grover's algorithm.
However, this is still exponential in n, which sucks.

- With the small protein shown earlier, trying a billion rounds a second would take 10^{31} times the age of the universe to run

Large Reconstruction

```
110100110
      011010000
            000000100
                  010010000
                        000000001
                              000100100
                                    010010010
                                          001001011
                                                101111100
-----------------------------------------------------------
110100110100000010010000000010010010010010101111100
```

It is feasible to determine small chunks of structure and then stitch the chunks together.

Large Reconstruction

Piecing together the chunks gets almost the right answer, with a few ambiguities.

0010?1001011111101101111111101101101101101000001111010001010000011011011111101111110110110110111111011111110000011?11110110010?10?10010011010001010010010110001111010000?10110111011111111110010010000010010110000110010010111011000110010?110110110101?11001101010111111011?11011011011111111101111110011100011110110110111011011111111111000111001011011011111010110?01110

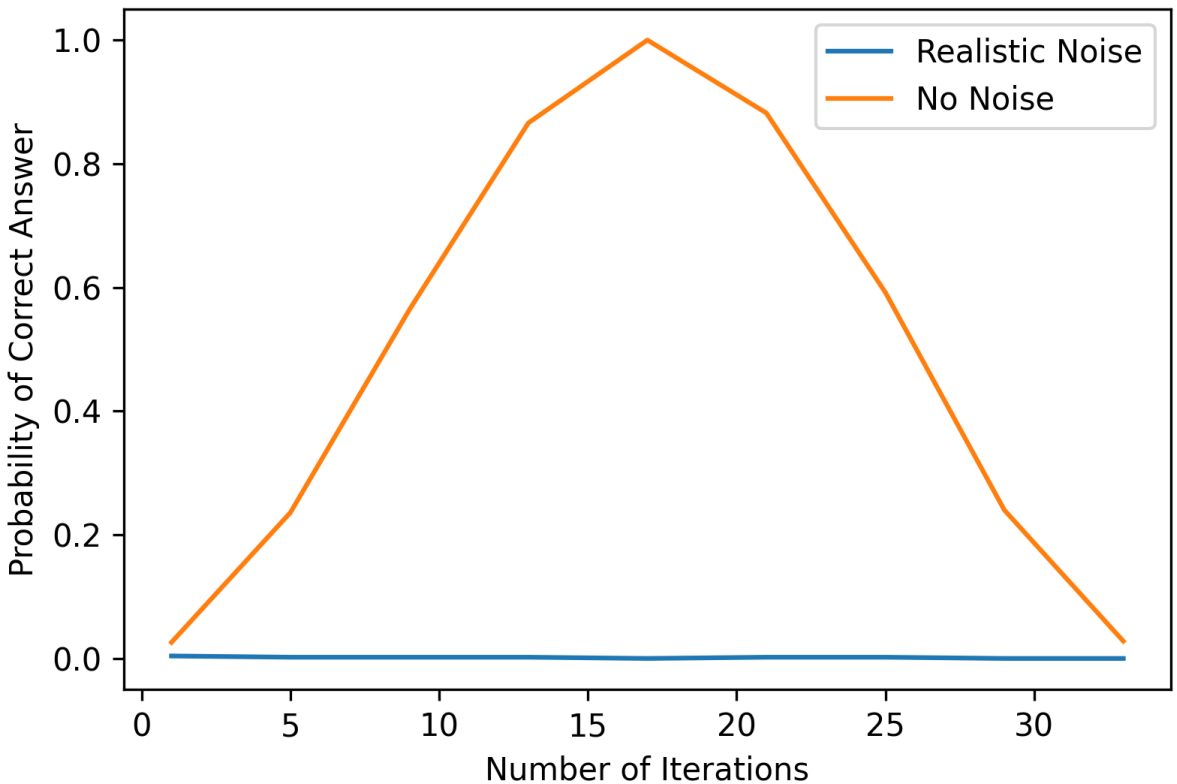
However, with only 9 ambiguities, this takes $\frac{\pi}{4}2^{9/2} = 18$ attempts to fill in (compared to $2^9 = 512$ with a classical computer), so we can use Grover's again!

Computer Noise

I took the noise model from `IBMQ_16_Melbourne`, the largest machine I have access to and ran the circuit with it.
The result:

Computer Noise

I took the noise model from `IBMQ_16_Melbourne`, the largest machine I have access to and ran the circuit with it.
The result: absolute trash

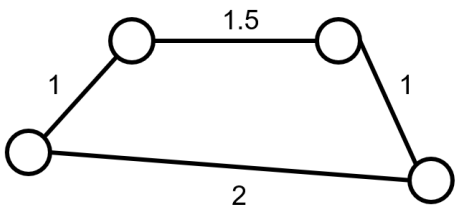


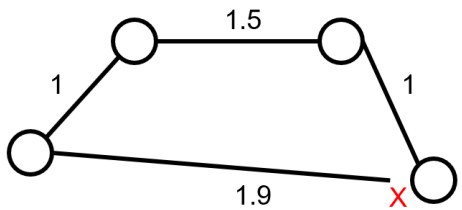
Data Noise

I added a tiny amount of noise, as low as a normal distribution with $\sigma = 0.01$ to the measured distances.
The result:

Data Noise

I added a tiny amount of noise, as low as a normal distribution with $\sigma = 0.01$ to the measured distances.
The result: absolute trash





Analysis

	Classical Version	Quantum Version
Problem Applicability		
Runtime		
Memory		
Accuracy		
Computation		
Noise Robustness		
Data		
Noise Robustness		

Analysis

	Classical Version	Quantum Version
Problem Applicability	Good for proteins and other biopolymers with sufficient measurement capability	
Runtime		
Memory		
Accuracy		
Computation		
Noise Robustness		
Data		
Noise Robustness		

Anything where the atoms form a chain and you can get distances between every four atoms is good.

Analysis

	Classical Version	Quantum Version
Problem Applicability	Good for proteins and other biopolymers with sufficient measurement capability	
Runtime	$O(n^2 2^n)$	
Memory		
Accuracy		
Computation		
Noise Robustness		
Data		
Noise Robustness		

There are 2^n possible configurations, and for each configuration it takes $O(n^2)$ to check if all the pairwise distances are good.

Analysis

	Classical Version	Quantum Version
Problem Applicability	Good for proteins and other biopolymers with sufficient measurement capability	
Runtime	$O(n^2 2^n)$	$O(n^2 2^{n/2})$
Memory		
Accuracy		
Computation		
Noise Robustness		
Data		
Noise Robustness		

The oracle still takes $O(n^2)$ time to run. The diffusion operator takes $O(n)$ time to run. Both operators are called $O(2^{n/2})$ times.

Analysis

	Classical Version	Quantum Version
Problem Applicability	Good for proteins and other biopolymers with sufficient measurement capability	
Runtime	$O(n^2 2^n)$	$O(n^2 2^{n/2})$
Memory	$O(n^2)$	
Accuracy		
Computation		
Noise Robustness		
Data		
Noise Robustness		

There are $O(n^2)$ pairwise distances that must be compared to.

Analysis

	Classical Version	Quantum Version
Problem Applicability	Good for proteins and other biopolymers with sufficient measurement capability	
Runtime	$O(n^2 2^n)$	$O(n^2 2^{n/2})$
Memory	$O(n^2)$	
Accuracy	Perfect	
Computation		
Noise Robustness		
Data		
Noise Robustness		

Assuming no noise, both versions can perfectly recreate the original structure

Analysis

	Classical Version	Quantum Version
Problem Applicability	Good for proteins and other biopolymers with sufficient measurement capability	
Runtime	$O(n^2 2^n)$	$O(n^2 2^{n/2})$
Memory	$O(n^2)$	
Accuracy	Perfect	
Computation	Perfect	
Noise Robustness		
Data		
Noise Robustness		

Classical computers are extremely robust to errors in computation by now.

Analysis

	Classical Version	Quantum Version
Problem Applicability	Good for proteins and other biopolymers with sufficient measurement capability	
Runtime	$O(n^2 2^n)$	$O(n^2 2^{n/2})$
Memory	$O(n^2)$	
Accuracy	Perfect	
Computation	Perfect	Complete Failure
Noise Robustness Data		
Noise Robustness		

Quantum computers are very noisy and this algorithm is much too sensitive to the noise.

Analysis

	Classical Version	Quantum Version
Problem Applicability	Good for proteins and other biopolymers with sufficient measurement capability	
Runtime	$O(n^2 2^n)$	$O(n^2 2^{n/2})$
Memory	$O(n^2)$	
Accuracy	Perfect	
Computation Noise Robustness	Perfect	Complete Failure
Data Noise Robustness	Complete Failure	

The algorithm can not tolerate any errors in the input data

Final Conclusions

1. Theoretically, this algorithm works, and Grover's algorithm gives a large $2^{n/2}$ speedup.

Final Conclusions

1. Theoretically, this algorithm works, and Grover's algorithm gives a large $2^{n/2}$ speedup.
2. However, this is still exponential
   - Does not help with $P \stackrel{?}{=} NP$

Final Conclusions

1. Theoretically, this algorithm works, and Grover's algorithm gives a large $2^{n/2}$ speedup.
2. However, this is still exponential
   - Does not help with $P \stackrel{?}{=} NP$
3. Algorithm cannot handle noise at all

Final Verdict

Final Conclusions

1. Theoretically, this algorithm works, and Grover's algorithm gives a large $2^{n/2}$ speedup.
2. However, this is still exponential
   - Does not help with $P \stackrel{?}{=} NP$
3. Algorithm cannot handle noise at all

Final Verdict

Useless

Sources

- Aaronson, S. Introduction to Quantum Information Science, Lecture 22. `https://www.scottaaronson.com/qclec/22.pdf` (2019).
- Abraham, A et al. Learn Quantum Computation Using Qiskit. (2020).
- Abraham, H. et al. Qiskit: An Open-source Framework for Quantum Computing. (2019).
- Boyer, M., Brassard, G., Hoeyer, P. & Tapp, A. Tight bounds on quantum searching. `arXiv:quant-ph/9605034` (1996)
- Lavor, C., Liberti, L. & Maculan, N. Grovers Algorithm applied to the Molecular Distance Geometry Problem. in Anais do 7. Congresso Brasileiro de Redes Neurais 14 (SBRN, 2016).
- Plotly Technologies Inc. Collaborative data science. https://plot.ly (2015).

Code stored at: `https://github.com/azswartz/Grovers-MDGP`