Reverse Engineering:

A Quantitative Analysis on the 7-Letter Scoring Scheme's Efficacy in mRNA Vaccine Designs

Junyou Li
University of Waterloo
Waterloo, Canada
simonjunou.li@gmail.com

***Abstract*—Although the COVID-19 vaccines are "95% effective" [1], they have to be stored in extreme temperatures. Moreover, certain vaccinated patients have shown signals of blood clots. Therefore, it is imperative to improve the stability and safety of the vaccine. However, there is no publicly available research on improving vaccine designs. Nevertheless, research in related fields can be applied to optimize vaccines. To optimize an mRNA sequence, one must minimize the free energy of the sequence (MFE) [2] and maximize the codon adaptation index (CAI) or the usage of optimal codons [3]. The Nussinov and Jacobson model (1980), implemented with the 7-letter scoring scheme suggested by Prof. Forbes J. Burkowski, was used to modify the Wuhan reference sequence. The result of the method was tested with the RNAfold WebServer and EMBOSS and compared with the Pfizer mRNA vaccine. The resulting sequence achieved a lower MFE than both the reference sequence and the Pfizer sequence. However, the method was less successful in maximizing CAI as the resulting sequence's CAI score was lower than Pfizer's but higher than that of the reference sequence.**

I. Introduction

. For the last year and a half, the world has lived within the terror of the SARS-SoV2 virus, commonly known as the novel coronavirus or COVID-19. To battle the pandemic, multiple pharmaceutical companies designed mRNA COVID-19 vaccines to help people gain an active immunity against the virus. Although the vaccines are "95% effective in preventing COVID-19" [1], they need to be stored in extreme temperatures. While the regular flu vaccines are stored at +2°C to +8°C, according to the Government of Canada (2015), the Ontario Ministry of Health (2021) suggests the mRNA COVID-19 vaccines to be stored between -80°C to -60°C. Moreover, certain vaccinated patients have shown signals of blood clots, raising concerns on whether the benefits of vaccination still outweigh the risks.

Under the current circumstances, it is imperative to explore if designing a more stable and safe vaccine is possible. However, currently, there is no publicly available research on vaccine designs. Nevertheless, there are techniques and models in related fields such as computer science and bioinformatics that can be applied to optimize the stability and safety of mRNA vaccines.

To optimize the stability and safety of an mRNA sequence, one must minimize the free energy of the sequence [2] and maximize the codon adaptation index or the usage of optimal codons [3]. Contrary to the lack of scholarly conversation on vaccine designs, there are multiple previous studies on modifying an RNA sequence. The models such as the Nussinov and Jacobson model (1980) and the LinearDesign (2020) model were proposed to attain the most stable RNA sequence [4, 5]. Furthermore, Prof. Forbes J. Burkowski, a bioinformatics Professor Emeritus at the University of Waterloo and the expert advisor of this study, suggested that the 7-letter scoring scheme can simplify the process of modifying the sequence, thus informing the research question: how could 7-Letter Scoring Scheme implemented in Nussinov and Jacobson model optimize the Minimum Free Energy and Codon Adaptation Index of mRNA COVID-19 vaccine designs?

To investigate the validity and effectiveness of the 7-letter scoring scheme in designing vaccines, a python bottom-up dynamic programming algorithm was written. The algorithm was given a COVID-19 spike protein RNA sequence which contained nucleotides A, C, G, and U and would be modified based on the 7-letter scoring scheme, the Nussinov and Jacobson model, and the goal of maximizing the optimal usage of codons, namely the optimal usage of nucleotide triplets. The RNAfold WebServer and EMBOSS were then used to calculate the modified sequence's minimum free energy and codon adaptation index, which were compared to those of the Pfizer mRNA COVID-19 vaccines. Based on the current mRNA COVID-19 vaccine's extreme storage temperature requirement and risks of health side-effects, the researcher hypothesized that the 7-letter scoring scheme would design a more stable and safer mRNA COVID-19 vaccine than the existing ones.

II. Literature Review

Vaccines, a biological preparation that offers immunity against a particular infectious disease, are often regarded as the most efficient means of contagious disease prevention [6]. Researchers have extensively studied vaccines – including vaccine allergies, effectiveness, hesitancy, manufacturing, and many other elements. However, no existing scholarly conversations surround how vaccines are designed, and, specifically, these design objectives are not publicly available.

This literature review aims to explore the scholarly conversations in related fields, such as computer science and bioinformatics, to inquire how existing knowledge could be implemented to develop an algorithm that designs a vaccine sequence and gain insights into the existing vaccine's design objectives by comparing the sequences

A. Algorithms and Approaches

a) Dynamic Programming

Dynamic Programming, or known as DP for short, is based on the idea that "a complex problem is decomposed into sub-problems and the solution to the complex problem is constructed from the solutions found for the sub-problems" [7]. This process could be repeatedly applied to subproblems as well. In other words, dynamic programming

is an algorithmic technique that leverages previous computation to save time.

The concept of DP was formulated during the 1940s, while the first paper was published in 1952 by American mathematician Professor Richard Bellman [8]. Dr. Bellman realized that DP could be used to obtain numerical solutions to deterministic optimization problems, in which the outputs are solely dependent on the parameters and initial conditions of the problem [9]. After attempting enumeration and various other methods, Bellman discovered that the same technique was used repeatedly to obtain the solution and has come up with a meta-strategy as the following. Firstly, the complex problem is generalized by transforming or decomposing it into a family of similar subproblems [7]. After a functional equation "relating the solution of each of these subproblems to the solutions of the others" is found, it is used to solve a subproblem [7]. Lastly, the solution of one subproblem is used to recover the solution of another particular subproblem [7].

b) Greedy Algorithm

A greedy algorithm is an algorithm that always takes the best local or immediate, optimal solution of subproblems while finding an answer for the overall problem. The algorithm is, according to Zhang et al. [10], "can be much faster than traditional dynamic programming approaches." The reason is that since greedy algorithm only explores locally optimal solutions, many calculations are skipped, while dynamic programming algorithm calculates and inspects all possible values [10].

However, computer scientists have raised doubts on the greedy algorithm's efficacy in finding the absolute optimal solution. The greedy algorithm is considered a combinatorial optimization, meaning that it relates the optimal solutions of a selection of arguments from a larger group without considering the effect of the choices made on the larger group. Bang-Jensen et al. state that because of these characteristics, the greedy algorithm "rarely outputs optimal solutions" and often "provides some kind of 'approximation' " to the actual optimal solution [11]. Feige also testifies this statement as he claims that greedy algorithms are known to produce suboptimal results on many problems [12].

From the above studies, dynamic programming is concluded as an optimization algorithm that considers all possible cases and returns the most optimal solution, while the greedy algorithm only takes local optimal solutions to find the overall optimal solution. Although the greedy algorithm provides an approximation of optimal solution faster than DP, considering the study aims to design the most stable and safe mRNA vaccine, DP is more ideal as it returns an absolute optimal solution. Therefore, this study has chosen dynamic programming as the algorithm framework.

c) DP – Top-Down vs. Bottom-Up

The dynamic programming meta-strategy described above only serves as a framework for dynamic programming. Within that framework, there are various mechanisms and techniques that focus on specific objectives. The following section introduces the two most common DP mechanisms, top-down and bottom-up, that are applicable to the goal of this study.

Top-down DP utilizes a technique called recursion. The idea of recursion comes from the verb "recur," which means "to return to a place or status" [13]. The mathematical interpretation of recursion is defined as a function f at an argument x which repeatedly calls itself and divides x into subproblem until x becomes the base case [13]. To put it another way, top-down DP takes in the desired answer's argument and recursively calls itself until the base case is met. Memoization, "a mechanism that is separate from the original program to save the result of each function call" [14], is commonly used in top-down DP to reduce the execution time [15-17]. The idea of memoization is to, instead of immediately compute the function itself, "modify recursive calls to first look up in the table, and then, if the subproblem has been computed, use the saved result, otherwise, compute it and save the result in the table" [14].
Bottom-up DP, on the other hand, utilizes a technique called tabulation. Tabulation computes in a bottom-up fashion that the solution to the complex super question is computed using the available subproblem solutions [18-20]. This mechanism overcomes the disadvantages of memoization by combining the mechanism with the resulting program and, most importantly, eliminate the use of recursion [14]. However, it has its own disadvantages. For example, it often requires a thorough understanding of the question to rewrite the program, and since it statically ensures all possible values are calculated, the table used is often larger than necessary [21].

To compare the efficacy of the approaches, Dixit et al. pointed out that "tabulation is faster as compared to memoization because we directly have access to the past values from the data structure, while memoization is slow due to recursive calls" [22]. However, memoization requires much less space than tabulation, as it does not need to "solve every subproblem to arrive at a desired case" [22].

Bottom-up dynamic programming, from the above studies, is concluded as a superior DP mechanism when occupying a considerable among of memory is not a concern. Considering that the length of the sequence would be 3822 nucleotides, this study has chosen bottom-up DP and tabulation to be the DP mechanism as it is much more time-efficient when processing large arguments while occupying a considerable amount of memory is not an issue.

B. Secondary Structure & RNA

Since the objective of this study is to evaluate the efficacy of the 7-letter scoring scheme, it requires a model algorithm to implement the scoring scheme. Thus, the following paragraphs discuss several successful algorithms that find the most stable secondary structure of an RNA sequence.
The Nussinov and Jacobson model is a computer method developed by Ruth Nussinov and Ann B. Jacobson to find the most stable secondary structures in long single-stranded RNAs [4]. The essence of the algorithm is the idea of

fragments and base-pairings. A fragment can be defined as a sequence of consecutive nucleotides within an RNA sequence, and a base-pairing is when two nucleotides are bound to each other by hydrogen bonds, they form a base pair, which is either AU, CG, or GU. Most importantly, the RNA sequence with minimum free energy would be the one in which all the non-overlapping fragments have the lowest possible free energy [4]. For instance, if an RNA sequence could only be divided into two non-overlapping fragments, the RNA sequence with the most hydrogen bonds is the one where both fragments have maximum hydrogen bonds.

After various testing, these results presented a strong argument that the model is "both simpler and faster than other existing procedures," as their model was about 100 times faster than preexisting algorithms [4]. The results also gave credence to the model's ability to produce the best secondary structure, as its result only had differences, compared to the published structure, that were minor in nature and was "virtually identical with the published structure" [4].

Although Nussinov and Jacobson model was able to accurately find an optimal secondary structure by obtaining maximum base-pairings, it cannot be modified to compute the stacking energy, which accounts for free energy gained from stacked base pairs [23]. Therefore, Zuker et al. designed an dynamic programming algorithm what is in essence a fusion of the Nussinov and Jacobson model and stacking energies [24]. The results of the algorithm presented evidence that the Zuker model was able to find a more stable secondary structure than previous algorithms, as its computed structure has a 15% improvement in minimum free energy [24]. In addition, the implementation of stacking energy "eliminated much tedious work," resulting in a much simpler and faster algorithm [24]. However, there were limitations to the study as Zuker et al. pointed out that the applied thermodynamic rules were inadequate and would not yield "a biologically meaningful folding" [24].

Zhang et al. [5], in a more recent study on LinearDesign, explored the novel idea of using a Deterministic Finite Automaton (DFA) to represent all the synonymous codons [5]. The application of DFA to represent synonymous codons allows the LinearDesign program to modify the sequence, as it provides different nucleotide options at each index to optimize base pairing and free energy. Zhang et al. also implemented a technique called "beam pruning" in order to achieve a linear time complexity, meaning that the execution time will increase linearly as the argument increases [5].

The results reveal that the implementation of beam pruning and DFA is able to achieve a linear time complexity, which is much faster than the cubic time complexity of the Nussinov and Jacobson model and the Zuker model[5]. Furthermore, the LinearDesign script was able to finish the entire spike protein of the SARS-CoV-2 in under eleven minutes, while it would take the other two models more than an hour to finish [5]. Despite the unreal speed, the LinearDesign also demonstrates promising abilities to minimize the MFE as its sequence's free energy difference with the exact value is only 0.6% [5].

To summarize, the algorithms from the above studies have their own advantages and disadvantages. The Nussinov and Jacobson was time-efficient and accurate but did not account for stacking energies. On the other hand, although the Zuker model took stacking energies into account, the model was complicated to implement and had certain limitations. Lastly, the LinearDesign was incredibly fast, but the implementation of the beam pruning was sophisticated and required a high-performance computer. Even though all of them are widely accepted modelled that are utilized in various fields, the Nussinov and Jacobson model stands out from them all. Since the objective of this study is to evaluate the efficacy of the 7-letter scoring scheme in designing mRNA vaccines, the Nussinov and Jacobson model fits the study the most as the 7-letter scoring scheme can be realized without major modifications in the algorithm. Therefore, the study chose the Nussinov and Jacobson model as the algorithm to implement the 7-letter scoring scheme.

III. Methodology

A. Study Design

In order to fully understand the 7-letter scoring scheme's efficacy in designing mRNA COVID-19 vaccines, a non-experimental quantitative design-based study was conducted. Implementing a quantitative method allows results to be evaluated on the algorithm performance by comparing and contrasting to the existing vaccines. As such, this study utilizes two standardized measures to analyze the efficacy of the design, where

- stability of the RNA sequence secondary structure was evaluated based on the minimum free energy (MFE) of the RNA sequence;
- safety of the vaccine was evaluated based on the Codon Adaptation Index (CAI) proposed by Sharp et al. [25] to measure the synonymous codon usage bias, which is useful for "assessing the adaptation of viral genes to their hosts" [25].

Since the algorithm dealt with the entire spike protein sequence, which is 3,822 nucleotides long, the bottom-up DP and tabulation were selected as the dynamic programming mechanism since they are more efficient than top-down DP and memoization [22].

B. Sample Description

Vaccine designs and their sequences have often been kept secret from other pharmaceutical companies as they exhibit how one's vaccine is designed. Thus, only the mRNA COVID-19 vaccine sequence from Pfizer Inc. was found to be publicly available.

NCBI, National Center for Biotechnology Information, provides a platform for certified research centers and researchers to upload RNA sequencing results for the SARS-CoV2 virus. From over 430,000 available viral sequences, the first-gen Wuhan viral sequence was chosen as the reference sequence. This allowed the algorithm to design a vaccine sequence using the same reference sequence as

Pfizer. When comparing the reference sequence to the Pfizer sequence, the researcher found out that Pfizer has replaced two amino acids with proline. To eliminate confounding variables, the amino acids at the same location in the reference sequence were also replaced with proline.

C. 7-Letter Scoring Scheme

Traditionally, an RNA sequence is comprised of four nucleotides, A, C, G, and U, and they could be classified as either R or Y, which are short for a purine (A or G) and a pyrimidine (C or U) respectively. Within each group, the nucleotides are structurally similar and thus can often substitute each other [26], forming synonymous codons. Take tyrosine as an example again. Tyrosine can be translated from UAU and UAC. Since both U and C are pyrimidines, the codon for tyrosine could be rewritten as UAY. However, it is possible that a nucleotide in a synonymous codon can be substituted by all four nucleotides. In this case, N, which stands for any nucleotide, is used to substitute the nucleotide. Proline, for example, can be translated from four different codons: CCU, CCC, CCA, and CCG. Since the last nucleotide can be any nucleotide, proline is rewritten as CCN. After adding the three letters - R, Y, and N - to the original four, there are 7 letters in total, hence the name 7-letter.

7-Letter Scoring Scheme is based on the idea that the two nucleotides in a base pair are bound to each other by hydrogen bonds. For example, in a GC pairing, there are three hydrogen bonds, while in a GU pairing, there are only two hydrogen bonds. Since hydrogen bonds play a critical role in stabilizing an RNA secondary structure [27], the 7-letter scoring scheme aims to maximize the number of hydrogen bonds. When there are synonymous codons, the scoring scheme prefers the codon with more hydrogen bonds to increase the stability of the sequence. Thus, in a GY pairing, Y becomes C as the GC pairing has more hydrogen bonding than GU pairing.

D. Procedure

Figure 1 represents the 2D tabulation table, where i represents the row, and j represents the column. The main diagonal is the mRNA viral spike protein sequence that is being modified. The sequence in the table is not the complete SARS-CoV2 spike protein sequence, as it is only a representation of what was done to it. The following three diagonals above the main diagonal are crossed off, known as the forbidden zone. In the simplest terms, this phenomenon happens because to form a base pair, two nucleotides must be at least three nucleotides apart. In other words, a nucleotide can only form a base pair starting from the fourth nucleotide away from it. Thus, the first three diagonals are the forbidden zone that prohibits base pairings.

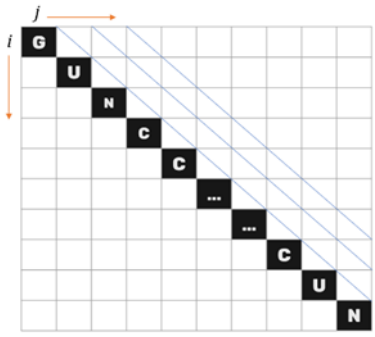

Figure 1 A graphical representation of the initial 2D table in bottom-up DP

Furthermore, the Nussinov and Jacobson model was chosen as it was easy to implement and fit seamlessly with the 7-letter scoring scheme. The main algorithm could be represented as below:

$$E(i,j) = min \begin{cases} E(i+1,j-1) + e(r_i, r_j) \\ E(i+1,j) \\ E(i,j-1) \\ \min\limits_{i+4 \le k \le j-5} \{E(i,k) + E(k+1,j)\} \end{cases} \quad [16]$$

where *i* and *j* represent the starting and ending index of the fragment. The *E(i,j)* function points to the box at row *i* and column *j* in the table and represents the maximum amount of hydrogen bond in the fragment starting from the i^{th} nucleotide to the j^{th} nucleotide. *e(ri,rj),* on the other hand, returns the number of hydrogen bonds between the base-pairing of nucleotides *ri* and *rj* using the 7-letter scoring scheme. To understand the logic of the algorithm, the first case that would be processed, *E(0,4),* was taken as the example.

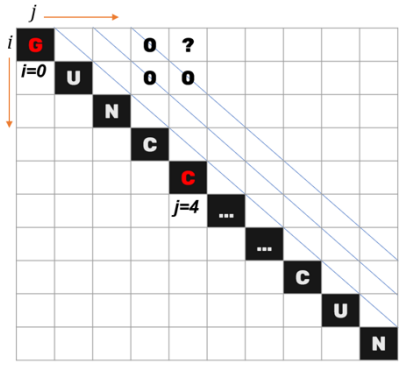

Figure 2 A graphical representation of cases in E(0,4)

In the first case, the maximum number of the hydrogen bond is equivalent to the sum of fragment *E(i+1,j-1)* and the base-pairing of the first and last nucleotides, G and C. Since *E(1,3)* is in the forbidden zone, it returns 0, while *e(G,C)* produces a value of -3. Case two and case three are similar, as both fragments are in the forbidden zone, returning a value of 0. Case four, bifurcation, has a precondition of $i+4 \le k \le j-5$, where *k* is the intersecting point of two non-overlapping segments. However, in this scenario when $i = 0$ and $j = 4$, the precondition is not met as zero plus four is larger than four minus five. Therefore, case four also has the value of 0. Looking at the values of all four cases, case one has the lowest value of -3. Thus, the value of *E(0,4)* is -3.

While the box *E(i,j)* is given the value -3, the inverted box in the lower triangle *E(j,i)* is given the value of 1 since *E(i,j)* is obtained through case 1. This step was called backtracing, which would modify the sequence after the table is filled up. The table was then being filled diagonal by diagonal.

After the table was filled up, the backtracing section began. Starting with *E(3822, 0),* which was responsible for recording the last case code, the program backtraced according to the case codes to find out what was done to obtain the most stable sequence. When the case code was one, the program would record the nucleotides involved in base-pairing according to the *i* and *j* value, converting 7-letter nucleotides back to conventional nucleotides if there were any. In the upper triangle, since case one is attained through fragment *E(i-1,j+1)*, the case code corresponding to this fragment is *E(j+1,i-1)*. Thus, when the case code is one and is at location *E(i,j)*, the case code prior is in *E(i+1,j-1)*, as the operation is inverted for finding the fragment of case one. For the other three cases, the same process was done, which was checking the previous box using the inverted procedure of the main algorithm.

There were still leftover 7-letter nucleotides in the sequence after backtracing as not all of them were involved in base-pairing. The program then replaced codons containing 7-letter nucleotides with conventional first rank codons to improve the CAI.

E. Collection Instruments

Python was chosen as the programming language as it provides various packages, such as NumPy, that allow programmers to deal with large matrices or tables efficiently. As the entire sequence length was 3,822, the table had a size of over 14 million, making using efficient tables imperative. PyCharm was the editor used in this study as it provides the most build-in features and a robust debugging system, which made programming much more straightforward.

The result was submitted to the RNAfold WebServer by the University of Vienna to calculate MFE and EMBOSS, a python bioinformatics package, to calculate CAI. RNAfold WebServer was chosen as the MFE calculator since it utilizes the MFE prediction model proposed by Mathews et al., which is the most widely accepted and accurate model [28].

IV. Results

The sequence designed by the algorithm was analyzed to determine the efficacy of the 7-letter scoring scheme in designing an mRNA COVID-19 vaccine. The analysis focuses on comparing the 7-letter modified sequence and Pfizer sequence using two standardized measurements: the Minimum Free Energy and the Codon Adaptation Index.

	Minimum Free Energy (kcal/mol)	Codon Adaptation Index	Similarity with Pfizer Vaccine
Wuhan Reference Sequence	-1072.20	0.646	72.18%
Pfizer Vaccine Sequence	-1321.30	0.944	100%
7-Letter Modified Vaccine Sequence	-1708.60	0.829	83.82%

Figure 4 Summary of the MFE, CAI, and similarity with Pfizer vaccine.

The results illustrate that the 7-letter scoring scheme was able to decrease the MFE by comparing to the Wuhan Reference Sequence and even managed to achieve a lower MFE than the Pfizer mRNA COVID-19 vaccine. Despite the success in minimizing MFE, the design was less successful in maximizing CAI, as it produced a CAI of 0.829, which is higher than the reference sequence but not the Pfizer vaccine sequence. Furthermore, the modifications 7-letter scoring scheme done to the original Wuhan reference sequence made it more similar to the Pfizer vaccine sequence as the then 72.18% similarity is now 83.82%. The 11.64% increase in similarity indicates that there are overlaps between the vaccine design objectives of this study and those of the Pfizer mRNA COVID-19 vaccine.

V. Discussion

The study was able to conclude that the implementation of the 7-letter scoring scheme is able to design an mRNA COVID-19 vaccine sequence with an optimized MFE and a good CAI. This finding is attested by a study conducted by Terai et al. [29] as it reached a virtually identical conclusion. Terai et al. developed a technique called CDS to find the most stable secondary structure by exploring all the possible synonymous codons [29]. This technique is similar to the 7-letter scoring scheme as both focus on the idea of finding the optimal solution within all possible synonymous codons. The only difference between these two is how they were designed and implemented. The CDS was then put into the Zuker model to assess its validity. Terai et al. were able to conclude that the CDS design was able to improve the model's efficiency and ability to find the most stable secondary structure as all the one hundred proteins used were more stable after being modified by the algorithm [29]. Thus, the findings by Terai et al. present a strong argument that it is not a surprise the 7-letter scoring scheme was able to design a more stable sequence.

In the LinearDesign article, Zhang et al., similar to this study, explores the idea of MFE and CAI joint optimization and did "a two-dimensional comparison, MFE and codon adaptation index (CAI), between mRNA sequences" [5]. Zhang et al. summarized the comparison as a graph, which can be seen in Figure 5.

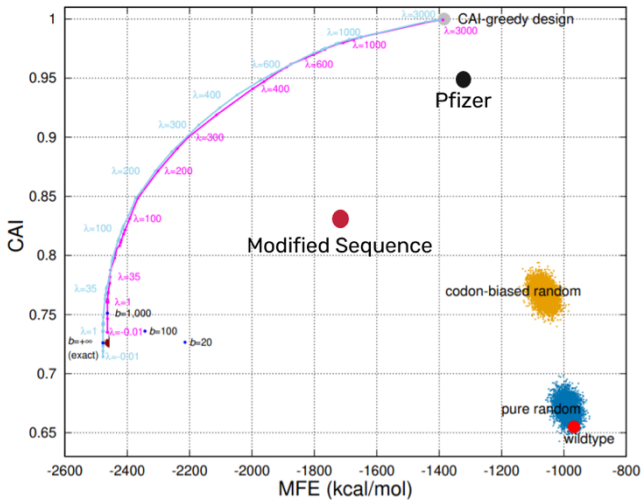

Figure 5 A graphical representation of the comparison between MFE and CAI of mRNA sequences.

In Figure 5, the x-axis represents the MFE in (kcal/mol), while the y-axis represents the CAI. Thus, the top left corner is the optimal spot as it has both the optimal MFE and optimal CAI. However, the top left corner where MFE is -2600 kcal/mol and CAI is 1 is not achievable as it's over the magenta curve. The magenta curve is "the accessible boundary of all possible sequences, i.e., no sequences can achieve the region beyond (to the top-left) the curve" [5]. To visualize where the modified sequence and Pfizer vaccine would be in this graph, two dots representing each sequence were plotted. The red dot for the modified sequence is located below the curve, indicating that the sequence is feasible. Comparing the two dots, it can be concluded that the 7-letter modified sequence has a more balanced design, while the Pfizer mRNA COVID-19 vaccine sequence is more CAI greedy. Nonetheless, the 0.829 CAI of the modified sequence, although not perfect in the human body, is considered adequate, as "CAI of >0.8 is regarded as good" [30].

VI. Conclusion

Understanding the different factors that affect the stability and safety of an mRNA COVID-19 vaccine is key to understanding how vaccines are designed. This study shows that theoretically, the implementation of the 7-letter scoring scheme is able to design a feasible mRNA sequence with an optimized MFE and a good CAI compared to the Pfizer mRNA COVID-19 vaccine. These findings have broader social and pharmaceutical implications that can help the public and researchers form a basic understanding of how the current vaccines might be designed. Moreover, researchers can build on this information and explore the feasibility and validity of other vaccine design objectives.

There are several limitations to this study that warrant discussion. First, the results are less generalizable due to the limited sample size. Since the Pfizer mRNA COVID-19 vaccine sequence was the only mRNA COVID-19 vaccine sequence available, extrapolating these findings to a definite conclusion would be erroneous. Furthermore, this study did not take pseudoknots into account. Yet, the existence of pseudoknots inhibits protein activities and stimulates -1 frameshifts [23]. In addition, although the free energy model used in the RNAfold WebServer was significantly predictive with a shorter sequence, the model becomes less accurate when the sequence length exceeds 700 nucleotides [28]. Considering the sequence length in this study was 3,822, it is reasonable to believe that the predicted MFE value can be imprecise. Lastly, the findings from this study are solely theoretical. Since all the evaluations made in this study are all based on theoretical models, it is unclear how the results will perform in real life as there are too many uncertain variables that could impact the design; thus, the efficacy of the 7-letter scoring scheme is only limited to modelling. In order to have a definite conclusion, future laboratory research and trials are required.

Even though the exact vaccine design objectives are still unclear, this study presents convincing evidence that optimizing MFE and CAI can be two of the many design objectives. The perception gained from this study can help form a basic model that assists future researchers to better approach the inquiry of vaccine designs. Undoubtedly, there are many more factors, such as minimizing stacking energy that accounts for free energy gained from stacked base pairs [27], which can be part of the vaccine design objectives. By gaining a better understanding of how vaccines are designed, the public can be better informed and potentially alleviate the negative image surrounding vaccines.

References

1 Polack, F.P., Thomas, S.J., Kitchin, N., Absalon, J., Gurtman, A., Lockhart, S., Perez, J.L., Pérez Marc, G., Moreira, E.D., and Zerbini, C.J.N.E.J.o.M.: 'Safety and efficacy of the BNT162b2 mRNA Covid-19 vaccine', 2020, 383, (27), pp. 2603-2615

2 Trotta, E.J.P.o.: 'On the normalization of the minimum free energy of RNAs by sequence length', 2014, 9, (11), pp. e113380

3 Mauger, D.M., Cabral, B.J., Presnyak, V., Su, S.V., Reid, D.W., Goodman, B., Link, K., Khatwani, N., Reynders, J., and Moore, M.J.J.P.o.t.N.A.o.S.: 'mRNA structure regulates protein expression through changes in functional half-life', 2019, 116, (48), pp. 24075-24083

4 Nussinov, R., and Jacobson, A.B.J.P.o.t.N.A.o.S.: 'Fast algorithm for predicting the secondary structure of single-stranded RNA', 1980, 77, (11), pp. 6309-6313

5 Zhang, H., Zhang, L., Li, Z., Liu, K., Liu, B., Mathews, D.H., and Huang, L.J.a.p.a.: 'LinearDesign: Efficient Algorithms for Optimized mRNA Sequence Design', 2020

6 Fiore, A.E., Bridges, C.B., and Cox, N.J.J.V.f.p.i.: 'Seasonal influenza vaccines', 2009, pp. 43-82

7 Sniedovich, M.: 'Dynamic programming' (CRC press, 1991. 1991)

8 Bellman, R.J.P.o.t.N.A.o.S.o.t.U.S.o.A.: 'On the theory of dynamic programming', 1952, 38, (8), pp. 716

9 Bellman, R., and Lee, E.J.I.C.S.M.: 'History and development of dynamic programming', 1984, 4, (4), pp. 24-28

10 Zhang, Z., Schwartz, S., Wagner, L., and Miller, W.J.J.o.C.b.: 'A greedy algorithm for aligning DNA sequences', 2000, 7, (1-2), pp. 203-214
11 Bang-Jensen, J., Gutin, G., and Yeo, A.J.D.o.: 'When the greedy algorithm fails', 2004, 1, (2), pp. 121-127
12 Feige, U.J.J.o.t.A.: 'A threshold of ln n for approximating set cover', 1998, 45, (4), pp. 634-652
13 Soare, R.I.J.B.o.s.L.: 'Computability and recursion', 1996, pp. 284-321
14 Liu, Y.A., and Stoller, S.D.: 'Dynamic programming via static incrementalization', in Editor (Ed.)^(Eds.): 'Book Dynamic programming via static incrementalization' (Springer, 1999, edn.), pp. 288-305
15 Michie, D.J.N.: '"Memo" functions and machine learning', 1968, 218, (5136), pp. 19-22
16 Friedman, D.P., Wise, D.S., and Wand, M.: 'Recursive programming through table look-up', in Editor (Ed.)^(Eds.): 'Book Recursive programming through table look-up' (1976, edn.), pp. 85-89
17 Hughes, J.: 'Lazy memo-functions', in Editor (Ed.)^(Eds.): 'Book Lazy memo-functions' (Springer, 1985, edn.), pp. 129-146
18 Bird, R.S.J.A.C.S.: 'Tabulation techniques for recursive programs', 1980, 12, (4), pp. 403-417
19 Cohen, N.H.J.A.T.o.P.L., and Systems: 'Eliminating Redundant Recursive Calls', 1983, 5, (3), pp. 265-299
20 Purdom Jr, P.W., and Brown, C.A.: 'The analysis of algorithms' (Holt, Rinehart & Winston, 1985. 1985)
21 Cormen, T.H., Leiserson, C.E., Rivest, R.L., and Stein, C.: 'Introduction to Algorithms' (McGraw-Hill, 2009. 2009)
22 Dixit, M., and Khan, M.A.: 'Analysis on Applicability and Feasibility of Dynamic Programming on the Basis of Time Complexity and Space Through Case Studies': 'Data, Engineering and Applications' (Springer, 2019), pp. 331-343
23 Giedroc, D.P., and Cornish, P.V.J.V.r.: 'Frameshifting RNA pseudoknots: structure and mechanism', 2009, 139, (2), pp. 193-208
24 Zuker, M., and Stiegler, P.J.N.a.r.: 'Optimal computer folding of large RNA sequences using thermodynamics and auxiliary information', 1981, 9, (1), pp. 133-148
25 Sharp, P.M., and Li, W.-H.J.N.a.r.: 'The codon adaptation index-a measure of directional synonymous codon usage bias, and its potential applications', 1987, 15, (3), pp. 1281-1295
26 Suzuki, D.T., and Griffiths, A.J.F.: 'An Introduction to Genetic Analysis' (W. H. Freeman, 1976. 1976)
27 Bordo, D., and Argos, P.J.J.o.m.b.: 'The role of side-chain hydrogen bonds in the formation and stabilization of secondary structure in soluble proteins', 1994, 243, (3), pp. 504-519
28 Mathews, D.H., Sabina, J., Zuker, M., and Turner, D.H.J.J.o.m.b.: 'Expanded sequence dependence of thermodynamic parameters improves prediction of RNA secondary structure', 1999, 288, (5), pp. 911-940
29 Terai, G., Kamegai, S., and Asai, K.J.B.: 'CDSfold: an algorithm for designing a protein-coding sequence with the most stable secondary structure', 2016, 32, (6), pp. 828-834
30 Huyan, T., Ruihua, T., Jing, L., Qi, L., Xiaoping, X., and Hui, Y.J.I.j.o.p.h.: 'Optimized expression and purification of humbug in Pichia pastoris and its monoclonal antibody preparation', 2015, 44, (12), pp. 1632