Balancing Minimum Free Energy and Codon Adaptation Index for Pareto Optimal RNA Design

Xinyu Gu
Department of Computer Science, University of Illinois Urbana-Champaign, Urbana, IL, USA

Yuanyuan Qi
Department of Computer Science, University of Illinois Urbana-Champaign, Urbana, IL, USA

Mohammed El-Kebir
Department of Computer Science, University of Illinois Urbana-Champaign, Urbana, IL, USA

Abstract

The problem of designing an RNA sequence $\mathbf{v}$ that encodes for a given target protein $\mathbf{w}$ plays an important role in messenger RNA (mRNA) vaccine design. Due to codon degeneracy, there exist exponentially many RNA sequences for a single target protein. These candidate RNA sequences may adopt different secondary structure conformations with varying minimum free energy (MFE), affecting their thermodynamic stability and consequently mRNA half-life. In addition, species-specific codon usage bias, as measured by the codon adaptation index (CAI), also plays an essential role in translation efficiency. While previous works have focused on optimizing either MFE or CAI, more recent works have shown the merits of optimizing both objectives. Importantly, there is a trade-off between MFE and CAI, i.e. optimizing one objective is at the expense of the other. Here, we formulate the Pareto Optimal RNA Design problem, seeking the set of Pareto optimal solutions for which no other solution exists that is better in terms of both MFE and CAI. We introduce DERNA (DEsign RNA), which uses the weighted sum method to enumerate the Pareto front by optimizing convex combinations of both objectives. DERNA uses dynamic programming to solve each convex combination in $\mathcal{O}(|\mathbf{w}|^3)$ time and $\mathcal{O}(|\mathbf{w}|^2)$ space. Compared to a previous approach that only optimizes MFE, we show on a benchmark dataset that DERNA obtains solutions with identical MFE but superior CAI. Additionally, we show that DERNA matches the performance in terms of solution quality of LinearDesign, a recent approach that similarly seeks to balance MFE and CAI. Finally, we demonstrate our method's potential for mRNA vaccine design using SARS-CoV-2 spike as the target protein.

2012 ACM Subject Classification Applied computing → Computational biology

Keywords and phrases Multi-objective optimization, dynamic programming, RNA sequence design, reverse translation, mRNA vaccine design

Digital Object Identifier 10.4230/LIPIcs.WABI.2023.21

Supplementary Material *Software (Source code)*: `https://github.com/elkebir-group/derna` archived at `swh:1:dir:d180327fb14fede1d3fcf822df273b4dc12c6069`

Funding *Mohammed El-Kebir*: National Science Foundation award number CCF 2046488

1 Introduction

With the emergence of the COVID-19 pandemic, messenger RNA (mRNA) vaccines have garnered significant attention due to their effectiveness in combating the disease [13, 17]. However, there remain significant challenges in the delivery [4] as well as the *in vitro* and *in vivo* stability of mRNA-based vaccines and therapeutics [28]. Importantly, due to codon degeneracy with $4^3 = 64$ codons encoding for 20 distinct amino acids as well as translation termination signals, there are exponentially many RNA sequences $\mathbf{v}$ for a single target protein $\mathbf{w}$. Synonymous codon choice impacts translational efficiency and mRNA stability in two interrelated ways. First, a subset of "optimal" codons occur at a higher frequency in


© Xinyu Gu, Yuanyuan Qi, and Mohammed El-Kebir;
licensed under Creative Commons License CC-BY 4.0

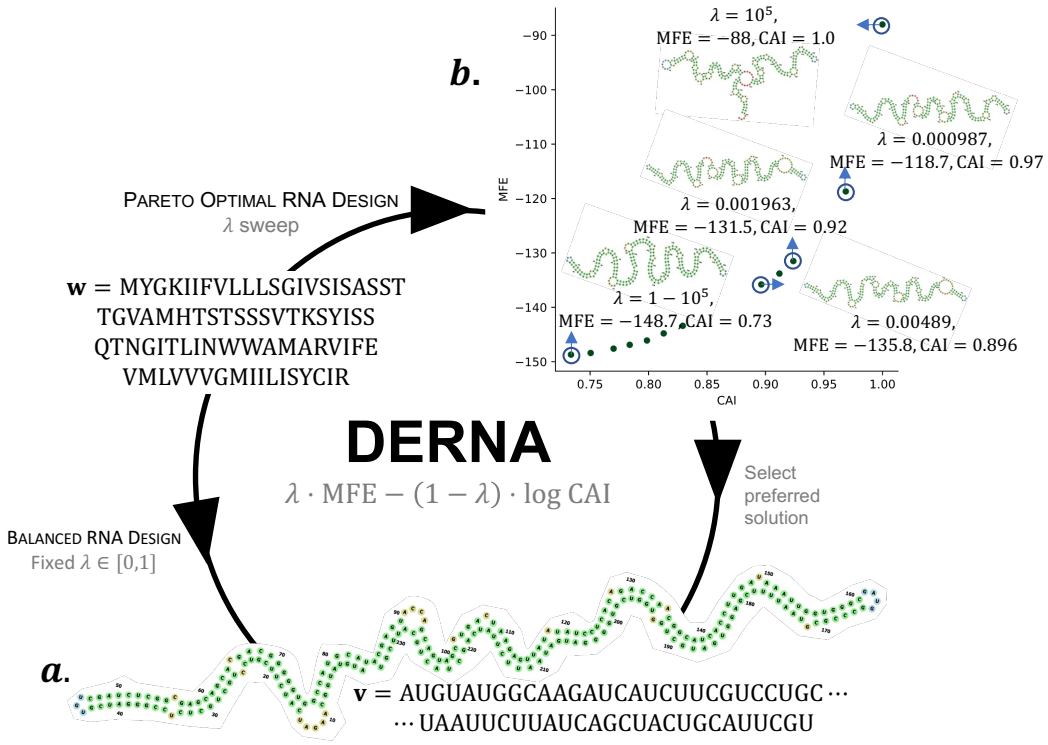

Figure 1 DERNA seeks Pareto optimal RNA sequences v for a target protein w, balancing the minimum free energy (MFE) and codon adaptation index (CAI). (a) For the BALANCED RNA DESIGN (BRD) problem, DERNA takes as input the parameter $\lambda \in [0, 1]$ and returns the RNA sequence $\mathbf{v}$ whose corresponding secondary structure P minimizes $\lambda \cdot \text{MFE}(\mathbf{v}, P) - (1 - \lambda) \cdot \overline{\text{CAI}}(\mathbf{v}, \mathbf{w})$. (b) For the PARETO OPTIMAL RNA DESIGN problem, DERNA performs a systematic sweep on λ, solving multiple BRD instances and returning a set of Pareto optimal solutions $(\mathbf{v}, P)$.

highly-expressed genes [8] and "non-optimal" codons lead to increased ribosomal pausing and decreased mRNA half-life [19, 29]. Second, depending on codon choice, each candidate RNA sequence folds into a distinct *secondary structure* or conformation, affecting its thermodynamic stability and consequently mRNA half-life. Recent studies have demonstrated the importance of both factors, showing that increased secondary structure as well as optimal codon usage lead to increased protein expression [16, 24]. This leads to the following key question of this paper. How does one identify RNA sequences that optimize both criteria?

Different organisms and even different genes within the same organism can have distinct codon usage patterns. The *codon adaptation index* (CAI) is a measure that quantifies the degree of codon usage bias in a protein coding sequence relative to a reference set of highly-expressed genes [22]. The reference set is often chosen based on the assumption that these genes have evolved to use codons that are mostly efficiently translated by the ribosome. Thus, an RNA sequence with high CAI is expected to have higher rates of translation [8, 19, 29]. Specifically, for a reference gene set, we are given the relative frequencies $g(\mathbf{x})$ of each codon $\mathbf{x}$ in the gene set. Then, the CAI of an RNA sequence $\mathbf{v}$ is the geometric mean of the ratios $g(\mathbf{x}) / \max_{\mathbf{y} \in S(\mathbf{x})} g(\mathbf{y})$ of each codon $\mathbf{x}$ vs. the maximum relative frequency of a synonymous codon $\mathbf{y} \in S(\mathbf{x})$ (see Equation (1)). RNA sequences that are composed of only optimal

codons with maximum relative frequencies have by definition a CAI of 1. In our setting, it is trivial to identify such an RNA sequence with CAI equal to 1 by simply choosing the codon with maximum relative frequency for each amino acid of the target protein. However, such an RNA sequence with optimal CAI may exhibit suboptimal amounts of secondary structure (Figure 1).

RNA molecules adopt secondary structure and three-dimensional conformations as the nucleotides within the RNA molecule and the surrounding solvent interact with each other. When an RNA molecule folds into its conformation, it forms base-pairing interactions between nucleotides that result in the lowest possible free energy [7]. This conformation is said to have the *minimum free energy* (MFE). In general, an RNA molecule with a lower MFE is more likely to be stable and maintain its integrity over time, whereas an RNA molecule with a higher MFE is more likely to be degraded. Thermodynamic stability is an important factor in identifying the most stable RNA sequences that are likely to be functional and efficient in producing a target protein [16, 24]. Zuker and Stiegler [32] introduced a dynamic programming algorithm to identify the conformation of RNA molecules with minimum free energy from a given RNA sequence $\mathbf{v}$. This approach was later extended independently by Terai et al. [23] and Cohen and Skiena [1] to identify a RNA sequence $\mathbf{v}$ and corresponding secondary structure with overall minimum MFE for a given target protein sequence $\mathbf{w}$. However, an RNA sequence with optimal MFE may have suboptimal CAI (Figure 1).

Recognizing the importance of examining both CAI and MFE, Zhang et al. [31] introduced LinearDesign, which uses stochastic context-free grammars and deterministic finite automata and applies a beam search heuristic to optimize $\text{MFE} + \lambda_{\text{LD}} \log \text{CAI}$ where λ_{LD} is a user-specified parameter.

In this work, we model the trade-off between CAI and MFE as a multi-objective optimization problem. That is, we introduce the Pareto Optimal RNA Design problem, seeking the set of *Pareto optimal solutions* for which no other solution exists that is better in terms of both MFE and CAI (Figure 1). We use the weighted sum method [30] to enumerate the Pareto front by optimizing convex combinations of both objectives – leading to the Balanced RNA Design problem (Figure 1). Our resulting algorithm, DERNA (DEsign RNA), extends the Zuker and Stiegler dynamic programming approach [32] to solve each convex combination in $\mathcal{O}(|\mathbf{w}|^3)$ time and $\mathcal{O}(|\mathbf{w}|^2)$ space. Unlike LinearDesign, where key functions are closed source, DERNA is fully open source with all code and functionality available to the user under a permissive license. We show on a benchmark dataset that DERNA obtains solutions with identical MFE but superior CAI compared to CDSfold [23]. Additionally, we show that DERNA matches LinearDesign's performance in terms of solution quality. Finally, we run our method on the SARS-CoV-2 spike protein and demonstrate its potential for mRNA vaccine design.

2 Problem Statement

A secondary structure for an RNA sequence with length n is a set of ordered base pairings $(i, j) \in [n] \times [n]$ such that each base is paired with at most one other base and there are no crossings base pairings (also known as pseudoknots). More formally, we define a secondary structure as follows.

▶ **Definition 1.** *A set $P \subseteq [n] \times [n]$ of base pairings is a* secondary structure *provided (i) for each base pairing $(i, j) \in P$ it holds that $i < j$, and for any two base pairings $(i, j), (i', j') \in P$ it holds that (ii) $i = i'$ if and only if $j = j'$ and (iii) if $i < i' < j$ then $i < i' < j' < j$.*

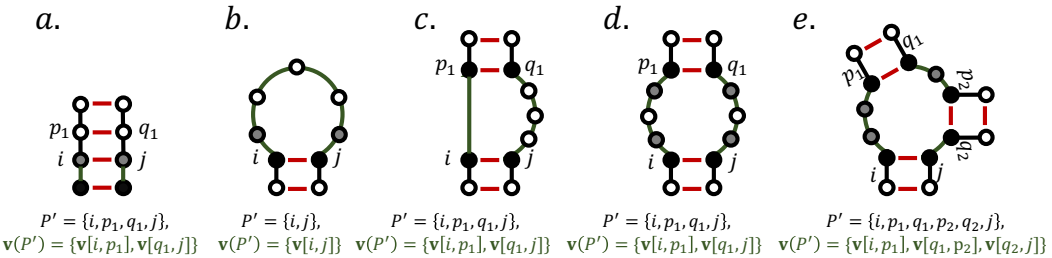

Figure 2 There are five secondary structure elements. (a) Stacking. (b) Hairpin loop. (c) Bulge loop. (d) Internal loop. (e) Multi-branch loop. Each structural element is defined by a unique set P' of nucleotide indices involved in base pairings (indicated in red). In addition, each structural element corresponds to a unique face of a planar embedding comprised of subsequences $\mathbf{v}(P')$ (indicated in green). Nucleotides next to the base pairings (indicated in gray) are involved in providing a free energy contribution to some structural components.

Following Zuker and Stiegler [32], a secondary structure P can be decomposed into several secondary structure elements, such that the free energy of the secondary structure P is the sum of the free energies contributed by each secondary structure element, defined as follows.

▶ **Definition 2.** *A subset* $P' = \{i, p_1, q_1, \ldots, p_k, q_k, j\} \subseteq [n]$ *of bases is a* secondary structure element *of* P *provided (i)* $i < p_1 < q_1 < \ldots < p_k < q_k < j$*, (ii)* $(i, j) \in P$*, (iii)* $(p_l, q_l) \in P$ *for each* $l \in [k]$ *and (iv) there exists no base pairing* $(i', j') \in P$ *such that* $i < i', j' < p_1 < j$*;* $i < q_k < i', j' < j$*; or* $q_l < i', j' < p_{l+1}$ *for all* $l \in \{1, \ldots, k-1\}$*.*

Alternatively, each secondary structure element corresponds to a unique face of a planar embedding of the secondary structure. Denoting the subsequence $v_i, \ldots, v_j$ by $\mathbf{v}[i, j]$, the subsequences that make up each face or secondary structure element are defined as follows.

▶ **Definition 3.** *A secondary structure element* $P' = \{i, p_1, q_1, \ldots, p_k, q_k, j\} \subseteq [n]$ *is comprised of* RNA subsequences $\mathbf{v}(P') = \{\mathbf{v}[i, p_1], \mathbf{v}[q_1, p_2], \ldots, \mathbf{v}[q_{k-1}, p_k], \mathbf{v}[q_k, j]\}$*. For* $k = 0$*, i.e.* $P' = \{i, j\}$*, the corresponding RNA subsequence* $\mathbf{v}(P')$ *equals* $\{\mathbf{v}[i, j]\}$*.*

Depending on the topology, we distinguish five types of secondary structure elements. In the simplest case, base pairing (i, j) is immediately followed by the pairing $(i+1, j-1)$, which is called a stacking element.

▶ **Definition 4.** *A secondary structure element* P' *of the form* $\{i, i+1, j-1, j\}$ *is a* stacking element*.*

Base pairing (i, j) forms a hairpin loop if there are no other base pairings involving bases $i+1, \ldots, j-1$.

▶ **Definition 5.** *A secondary structure element* P' *of the form* $\{i, j\}$ *is a* hairpin loop*.*

If base pairing (i, j) does not form a stacking element and there are other base pairings (i', j') occurring between i and j then (i, j) forms an interior loop element. We distinguish three types of interior loop elements: (i) a bulge loop, (ii) an internal loop and (iii) a multi-branch loop. The first two types correspond to a interior loop element enclosing a single base pairing (p_1, q_1), i.e. $k = 1$.

In a *bulge loop*, the enclosing base pairing is contiguous to either i or j, i.e. $p_1 = i + 1$ or $q_1 = j - 1$.

▶ **Definition 6.** *A secondary structure element P' of the form $\{i, p_1, q_1, j\}$ is a* bulge loop *provided (i) $(p_1, q_1) \neq (i+1, j-1)$ and (ii) $p_1 = i+1$ or $q_1 = j-1$.*

On the other hand, in an *internal loop* the enclosing base pairing is not contiguous, i.e. $p_1 > i+1$ and $q_1 < j-1$.

▶ **Definition 7.** *A secondary structure element P' of the form $\{i, p_1, q_1, j\}$ is an* internal loop *provided $i+1 < p_1 < q_1 < j-1$.*

If the interior loop element encloses more than one base pairing, i.e. $k > 1$, then we call the secondary structure element a *multi-branch loop*.

▶ **Definition 8.** *A secondary structure element P' of the form $\{i, p_1, q_1, \ldots, p_k, q_k, j\}$ is a* multi-branch loop *provided $k > 1$.*

As mentioned, we define the minimum free energy $\text{MFE}(\mathbf{v}, P) = \sum_{(i,j)\in P} \text{MFE}(\mathbf{v}, P, (i,j))$ of an RNA sequence $\mathbf{v}$ as the sum of the minimum free energies $\text{MFE}(\mathbf{v}, P, (i,j))$ of the secondary structure elements induced by each base pairing $(i,j) \in P$.

▶ **Definition 9.** *The* minimum free energy $\text{MFE}(\mathbf{v}, P)$ *of secondary structure P of RNA sequence $\mathbf{v}$ equals $\sum_{(i,j)\in P} \text{MFE}(\mathbf{v}, P, (i,j))$ where $\text{MFE}(\mathbf{v}, P, (i,j))$ is the contribution of the secondary structure element induced by a base pairing $(i,j) \in P$ defined as*

$$\text{MFE}(\mathbf{v}, P, (i,j)) = \begin{cases} f_{\text{s}}(\mathbf{v}(P')), & \text{if } P' = \{i, i+1, j-1, j\} \text{ is a stacking element,} \\ f_{\text{h}}(\mathbf{v}(P')), & \text{if } P' = \{i, j\} \text{ is a hairpin,} \\ f_{\text{b}}(\mathbf{v}(P')), & \text{if } P' = \{i, p_1, q_1, j\} \text{ is a bulge loop,} \\ f_{\text{i}}(\mathbf{v}(P')), & \text{if } P' = \{i, p_1, q_1, j\} \text{ is an internal loop,} \\ f_{\text{m}}(\mathbf{v}(P')), & \text{if } P' = \{i, p_1, q_1, \ldots, p_k, q_k, j\} \text{ is a multi-branch loop.} \end{cases}$$

The actual definitions of f_s, f_h, f_b, f_i and f_m depend on the used energy model. Briefly, in the widely used Turner energy model [26], the stacking energy value f_s is computed using a lookup table indexed by the four nucleotides comprising the base pairings $(i,j), (i+1, j-1)$. Similarly, the hairpin energy value f_h is a function of the four nucleotides $v_i, v_{i+1}, v_{j-1}, v_j$ and the length $j-i+1$ of the hairpin loop. For a bulge loop, the energy value f_b is a function of the four nucleotides in the base pairings $(i,j), (p_1, q_1)$ and the number of unpaired nucleotides in the loop $\mathbf{v}(\{i, p_1, q_1, j\})$. For an internal loop, the energy value f_h is a function of the eight nucleotides $v_i, v_{i+1}, v_{p_1-1}, v_{p_1}, v_{q_1}, v_{q_1+1}, v_{j-1}, v_j$ surrounding the base pairings $(i,j), (p_1, q_1)$ as well as the number of unpaired nucleotides in the loop $\mathbf{v}(\{i, p_1, q_1, j\})$. Finally, the energy value f_m is a function of the number k of base pairings enclosed in the multi-loop, the four nucleotides surrounding each base pairing and the number of unpaired nucleotides in the loop $\mathbf{v}(\{i, p_1, q_1, \ldots, p_k, q_k, j\})$. We refer to Appendix A.1 for more details.

The classical RNA Secondary Structure Prediction problem is defined as follows.

▶ **Problem 1** (RNA Secondary Structure Prediction (RSSP))**.** *Given an RNA sequence $\mathbf{v} \in \Sigma_{\text{rna}}^n$, find a secondary structure P such that $\text{MFE}(\mathbf{v}, P)$ is minimized.*

This problem can be solved in $O(n^3)$ time using the Zuker algorithm [32]. In this work we are interested in a reverse translation variant of the problem. That is, given a protein sequence $\mathbf{w} \in \Sigma_{\text{prot}}^m$ where Σ_{prot} is the set of 20 amino acids, we seek a corresponding RNA sequence $\mathbf{v} \in \Sigma_{\text{rna}}^{3m}$ that translates into $\mathbf{w}$. To that end, we use the function $S : \Sigma_{\text{prot}} \to \mathcal{P}(\Sigma_{\text{rna}}^3)$ such that $S(\alpha)$ is the set of codons that encode amino acid $\alpha \in \Sigma_{\text{prot}}$. We define $\sigma(a, s) = 3(a-1)+s$ to indicate the RNA sequence index corresponding to protein sequence index $a \in [m]$ and codon index $s \in \{1, 2, 3\}$.

▶ **Definition 10.** *RNA sequence* $\mathbf{v} \in \Sigma_{\text{rna}}^n$ encodes *for protein sequence* $\mathbf{w} \in \Sigma_{\text{prot}}^m$ *provided (i)* $|\mathbf{v}| = n = 3m = 3|\mathbf{w}|$ *and (ii)* $\mathbf{v}[\sigma(a,1), \sigma(a,3)] \in S(w_a)$ *for all protein indices* $a \in [m]$.

Rather than only considering the minimum free energy $\text{MFE}(\mathbf{v}, P)$, we also take species-specific codon usage bias into account. In other words, given species-specific relative codon frequencies $g : \Sigma_{\text{rna}}^3 \to [0,1]$, we compute the codon adaptation index $\text{CAI}(\mathbf{v}, \mathbf{w})$ defined as follows.

▶ **Definition 11.** *The* codon adaptation index $\text{CAI}(\mathbf{v}, \mathbf{w})$ *of RNA sequence* $\mathbf{v}$ *that translates into protein sequence* $\mathbf{w}$ *is defined as*

$$\text{CAI}(\mathbf{v}, \mathbf{w}) = \sqrt[m]{\prod_{a=1}^{m} \frac{g(\mathbf{v}[\sigma(a,1), \sigma(a,3)])}{\max_{\mathbf{x} \in S(w_a)} g(\mathbf{x})}} \tag{1}$$

where $g(\mathbf{x})$ *is the* species-specific relative frequency *of codon* $\mathbf{x} \in \Sigma_{\text{rna}}^3$ *such that* $g(\mathbf{x}) \geq 0$ *for all codons* $\mathbf{x}$ *and* $\sum_{\mathbf{x} \in \Sigma_{\text{rna}}^3} g(\mathbf{x}) = 1$.

The CAI ranges from 0 to 1, where a value of 1 indicates that for each amino acid w_a the maximum frequency codon $\arg\max_{\mathbf{x} \in S(w_a)} g(\mathbf{x})$ is used [22]. Thus, given a target protein sequence $\mathbf{w}$, there are two competing objective functions; we seek a corresponding RNA sequence $\mathbf{v}$ and secondary structure P that simultaneously minimizes $\text{MFE}(\mathbf{v}, P)$ and maximizes $\text{CAI}(\mathbf{v}, \mathbf{w})$. Equivalently, rather than maximizing $\text{CAI}(\mathbf{v}, \mathbf{w})$, we maximize $\overline{\text{CAI}}(\mathbf{v}, \mathbf{w})$ defined as

$$\text{CAI}(\mathbf{v}, \mathbf{w}) = \sqrt[m]{\prod_{a=1}^{m} \frac{g(\mathbf{v}[\sigma(a,1), \sigma(a,3)])}{\max_{\mathbf{x} \in S(w_a)} g(\mathbf{x})}} \propto \sum_{a=1}^{m} \log \frac{g(\mathbf{v}[\sigma(a,1), \sigma(a,3)])}{\max_{\mathbf{x} \in S(w_a)} g(\mathbf{x})} = \overline{\text{CAI}}(\mathbf{v}, \mathbf{w}). \tag{2}$$

We model the trade-off between MFE and CAI by introducing a parameter $\lambda \in [0,1]$ and minimizing a convex combination of $\text{MFE}(\mathbf{v}, P)$ and $-\overline{\text{CAI}}(\mathbf{v}, \mathbf{w})$.

▶ **Problem 2** (Balanced RNA Design (BRD)). *Given a protein sequence* $\mathbf{w} \in \Sigma_{\text{prot}}^m$ *and parameter* $\lambda \in [0,1]$*, find an RNA sequence* $\mathbf{v} \in \Sigma_{\text{rna}}^{3m}$ *with secondary structure* P *such that (i)* $\mathbf{v}$ *encodes for* $\mathbf{w}$ *and (ii) solution* $(\mathbf{v}, P)$ *minimizes* $\lambda \cdot \text{MFE}(\mathbf{v}, P) - (1-\lambda) \cdot \overline{\text{CAI}}(\mathbf{v}, \mathbf{w})$.

We say that a solution $(\mathbf{v}, P)$ is *Pareto optimal* if $(\mathbf{v}, P)$ is better than all other feasible solutions in at least one of the two objectives. In other words, there does not exist another solution $(\mathbf{v}', P')$ that is better in both objectives, or equal in one objective and better in the other. In our final problem, we seek all Pareto optimal RNA sequences $\mathbf{v}$.

▶ **Problem 3** (Pareto Optimal RNA Design (PORD)). *Given a protein sequence* $\mathbf{w} \in \Sigma_{\text{prot}}^m$*, enumerate all RNA sequences* $\mathbf{v} \in \Sigma_{\text{rna}}^{3m}$ *each with a secondary structure* P *such that (i)* $\mathbf{v}$ *encodes for* $\mathbf{w}$ *and (ii)* $(\mathbf{v}, P)$ *is Pareto optimal w.r.t. to* $\text{MFE}(\mathbf{v}, P)$ *and* $\text{CAI}(\mathbf{v}, \mathbf{w})$.

3 Methods

3.1 RNA Design with Fixed λ

In the Balanced RNA Design problem (Problem 2), we are given a protein sequence $\mathbf{w} \in \Sigma_{\text{prot}}^m$ and parameter $\lambda \in [0,1]$ that models the trade-off between MFE and CAI. In this section, we show how to solve this problem using dynamic programming. Specifically, for protein sequence indices $a, b \in [m]$, codon indices $s, t \in \{1,2,3\}$, codons $\mathbf{x} \in S(w_a)$ and

$\mathbf{y} \in S(w_b)$, $O[a][b][s][t][\mathbf{x}][\mathbf{y}]$ is the minimum objective value when solving a problem instance restricted to RNA sequence $\mathbf{v}[\sigma(a,s), \sigma(b,t)]$ such that codons $\mathbf{x}$ and $\mathbf{y}$ are used to encode amino acids w_a and w_b, respectively. Using $O[a][b][s][t][\mathbf{x}][\mathbf{y}]$, we express the objective value of an optimal solution as

$$\min_{\mathbf{x} \in S(w_1), \mathbf{y} \in S(w_m)} O[1][m][1][3][\mathbf{x}][\mathbf{y}]. \tag{3}$$

To see why this is the case, observe that $O[1][m][1][3][\mathbf{x}][\mathbf{y}]$ equals the minimum objective value for the complete RNA sequence $\mathbf{v}[\sigma(1,1), \sigma(m,3)] = \mathbf{v}[1, 3m] = \mathbf{v}$ restricted to using codons $\mathbf{x}$ and $\mathbf{y}$ for amino acid w_1 and w_m, respectively. Thus, the overall minimum objective value is obtained for the codon pair $(\mathbf{x}, \mathbf{y}) \in S(w_1) \times S(w_m)$ that minimizes $O[1][m][1][3][\mathbf{x}][\mathbf{y}]$.

Let $\Gamma = \{(A,U), (U,A), (G,C), (C,G), (G,U), (U,G)\}$ be the set of allowed base pairings in the Turner energy model [14]. To express the contribution of the CAI, we introduce the shorthand $\bar{g}(w, \mathbf{x}) = \log(g(\mathbf{x}) / \max_{\mathbf{y} \in S(w)} g(\mathbf{y}))$ such that $\overline{\text{CAI}}(\mathbf{v}, \mathbf{w}) = \sum_{a=1}^{m} \bar{g}(w_a, \mathbf{v}[\sigma(a,1), \sigma(a,3)])$. We define $O[a][b][s][t][\mathbf{x}][\mathbf{y}]$ recursively as

$$\min \begin{cases} -(1-\lambda)\bar{g}(w_a, \mathbf{x}), & \text{if } a = b, \mathbf{x} = \mathbf{y}, \\ \infty, & \text{if } a = b, \mathbf{x} \neq \mathbf{y}, \\ O[a][b][s+1][t][\mathbf{x}][\mathbf{y}], & \text{if } a < b, s \in \{1,2\}, \\ O[a][b][s][t-1][\mathbf{x}][\mathbf{y}], & \text{if } a < b, t \in \{2,3\}, \\ \min\limits_{\mathbf{x}' \in S(w_{a+1})} \{O[a+1][b][1][t][\mathbf{x}'][\mathbf{y}]\} - (1-\lambda)\bar{g}(w_a, \mathbf{x}), & \text{if } a \leq b-1, s = 3, \\ \min\limits_{\mathbf{y}' \in S(w_{b-1})} \{O[a][b-1][s][3][\mathbf{x}][\mathbf{y}']\} - (1-\lambda)\bar{g}(w_b, \mathbf{y}), & \text{if } a \leq b-1, t = 1, \\ \min\limits_{a \leq c < b, t' \in \{1,2\}, \mathbf{x}' \in S(w_c)} \left\{ \begin{array}{l} O[a][c][s][t'][\mathbf{x}][\mathbf{x}'] \\ \quad + E[c][b][t'+1][t][\mathbf{x}'][\mathbf{y}] \\ \quad + (1-\lambda)\bar{g}(w_c, \mathbf{x}') \end{array} \right\}, & \text{if } a < b, \\ \min\limits_{a \leq c < b-1, \mathbf{y}' \in S(w_c), \mathbf{x}' \in S(w_{c+1})} \left\{ \begin{array}{l} O[a][c][s][3][\mathbf{x}][\mathbf{y}'] \\ \quad + E[c+1][b][1][t][\mathbf{x}'][\mathbf{y}] \end{array} \right\}, & \text{if } a < b-1, \\ E[a][b][s][t][\mathbf{x}][\mathbf{y}], & \text{if } a < b, (x_s, y_t) \in \Gamma. \end{cases}$$

There are two components in the objective function, the CAI and the MFE. We account for MFE upon identifying structural elements at base pairing $(\sigma(a,s), \sigma(b,t))$ using the energy functions in Definition 9. To avoid double counting, we must ensure that CAI is only accounted for once for each codon. As such, we include a CAI contribution when crossing codon boundaries or reaching a valid base case.

The first case in the above recurrence corresponds to the base case where $a = b$ and $\mathbf{x} = \mathbf{y}$. In that case, base pairing between $\sigma(a,s)$ and $\sigma(b,t)$ is not possible as the Turner energy model [14] requires at least two nucleotides in between a pairing. In this base case, we must account for the CAI contribution of codon $\mathbf{x}$. The other base case occurs when $a = b$ and $\mathbf{x} \neq \mathbf{y}$, which is not allowed as any one amino acid must be encoded by a single codon – this case thus receives a value of ∞.

The next two cases correspond to, respectively, incrementing either the left index $\sigma(a,s)$ or decrementing the right index $\sigma(b,t)$ without crossing any codon boundary and leaving the corresponding nucleotide unpaired. As such, we do not have to account for CAI. However, in the following two cases, we additionally cross the codon boundary and thus must account for the CAI contribution of respectively codons $\mathbf{x}$ and $\mathbf{y}$. Next, we include two cases corresponding to bifurcating into two parts, one part is between nucleotides $\sigma(a,s)$ and

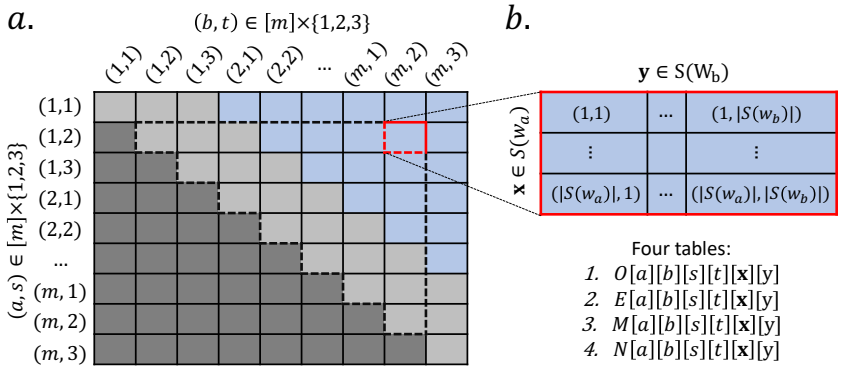

Figure 3 Dynamic programming for solving the Balanced RNA Design problem. (a) To solve this problem, we store four dynamic programming tables O, E, M and N with identical dimensions indexed as $[a][b][s][t][\mathbf{x}][\mathbf{y}]$. Rows and columns correspond to pairs $(a, s), (b, t) \in [m] \times \{1, 2, 3\}$, respectively, both ordered lexicographically in increasing order. With the exception of the base cases for table O where $a = b$ (indicated in light gray), the recurrences require $a < b$ (indicated in blue). The dashed lines outline the entries of the table on which the red entry depends. (b) Each entry $[(a, s)][(s, t)]$ expands into another codon-by-codon table, whose rows are codons $\mathbf{x} \in S(w_a)$ and columns are codons $\mathbf{y} \in S(w_b)$.

$\sigma(c, t')$ and the other part is between nucleotides $\sigma(c, t') + 1$ and $\sigma(b, t)$. In the first case, the split happens inside a codon, i.e. $t' \in \{1, 2\}$. We must include a correction of $+(1-\lambda)\bar{g}(w_c, \mathbf{x}')$ as both parts will include a CAI contribution of the same codon $\mathbf{x}'$. On other hand, when the split happens outside a codon, i.e. $t' = 3$ then no such correction is needed.

The last case corresponds to base pairing between $\sigma(a, s)$ and $\sigma(b, t)$. Specifically, $E[a][b][s][t][\mathbf{x}][\mathbf{y}]$ denotes the optimal objective value when nucleotides $v_{\sigma(a,s)}$ and $v_{\sigma(b,t)}$ correspond to codons $\mathbf{x}$ and $\mathbf{y}$, respectively, and form a base pairing. When calculating $E[a][b][s][t][\mathbf{x}][\mathbf{y}]$, we consider the minimum among the five cases corresponding the five secondary structures elements defined in Section 2. That is, $E[a][b][s][t][\mathbf{x}][\mathbf{y}]$ equals $\min\{E_s[a][b][s][t][\mathbf{x}][\mathbf{y}], E_h[a][b][s][t][\mathbf{x}][\mathbf{y}], E_b[a][b][s][t][\mathbf{x}][\mathbf{y}], E_i[a][b][s][t][\mathbf{x}][\mathbf{y}], E_m[a][b][s][t][\mathbf{x}][\mathbf{y}]\}$. The precise definitions are given in Appendix A.2. In particular, we require two additional recurrences $M[a][b][s][t][\mathbf{x}][\mathbf{y}]$ and $N[a][b][s][t][\mathbf{x}][\mathbf{y}]$ for solving the multi-branch loop case.

3.1.1 Dynamic Programming, Time and Space Complexity

We store the following four tables: (i) $O[a][b][s][t][\mathbf{x}][\mathbf{y}]$, (ii) $E[a][b][s][t][\mathbf{x}][\mathbf{y}]$, (iii) $M[a][b][s][t][\mathbf{x}][\mathbf{y}]$ and (iv) $N[a][b][s][t][\mathbf{x}][\mathbf{y}]$, each with the same dimensions. In particular, as each potential base pairing $(\sigma(a, s), \sigma(b, t))$ corresponds to exactly one of five structural elements, we do not store the corresponding values $E_s[a][b][s][t][\mathbf{x}][\mathbf{y}]$, $E_h[a][b][s][t][\mathbf{x}][\mathbf{y}]$, $E_b[a][b][s][t][\mathbf{x}][\mathbf{y}]$, $E_i[a][b][s][t][\mathbf{x}][\mathbf{y}]$ and $E_m[a][b][s][t][\mathbf{x}][\mathbf{y}]$ separately, but only their minimum value in $E[a][b][s][t][\mathbf{x}][\mathbf{y}]$. Note that the four stored tables have the same dimensions comprised of protein sequence indices $a, b \in [m]$, codon indices $s, t \in \{1, 2, 3\}$, and codons $\mathbf{x} \in S(w_a)$ and $\mathbf{y} \in S(w_b)$. Letting K denote the maximum number of codons associated with a single amino acid – the amino acids leucine (L), serine (S) and arginine (R) each have $K = 6$ of codons – we conclude that the space complexity is $\mathcal{O}(m^2K^2)$.

Inspection of the recurrences reveals that the computation of each entry $[a][b][s][t][\mathbf{x}][\mathbf{y}]$ in the four tables does not require access to entries $[a][b][s][t][\mathbf{x}'][\mathbf{y}']$ using other codons $\mathbf{x}' \neq \mathbf{x}$ and $\mathbf{y}' \neq \mathbf{y}$. On the other hand, we do require access to entries $[a'][b'][s'][t'][\mathbf{x}'][\mathbf{y}']$ where

$\sigma(a', s') \geq \sigma(a, s)$, $\sigma(b', t') \leq \sigma(b, t)$ (indicated with dashed lines in Figure 3). Moreover, with the exception of the base cases for table O, where $a = b$, the recurrences require $a < b$. This means we can organize the four tables as two-dimensional tables where the rows correspond to entries (a, s) and the columns correspond to entries (b, t), both sorted in increasing lexicographical order. Each entry $[(a, s)][(b, t)]$ corresponds to another two-dimensional table whose rows correspond to codons $\mathbf{x} \in S(w_a)$ and columns to codons $\mathbf{y} \in S(w_b)$ – see Figure 3b. We fill out the tables diagonally. More precisely, filling out the four entries indexed by $[a][b][s][t][\mathbf{x}][\mathbf{y}]$, we check if base pairing between $\sigma(a, s)$ and $\sigma(b, t)$ is possible, i.e. if $(x_s, y_t) \in \Gamma$. If so, we will first fill out the entry in E followed by N and then finally M. On the other hand, we will first fill out the entry in N, then M and finally E. After completely filling out tables E, M and N, we fill out table O. This ordering follows from the recurrences. We use back pointers to identify the optimal solution $(\mathbf{v}, P)$ when backtracing.

For each entry $O[a][b][s][t][\mathbf{x}][\mathbf{y}]$, the running time is dominated by the case to determine $E[a][b][s][t][\mathbf{x}][\mathbf{y}]$. That is, for each entry $E[a][b][s][t][\mathbf{x}][\mathbf{y}]$, it takes $\mathcal{O}(K^2)$ time to compute a stacking element or a hairpin loop element, $\mathcal{O}(mK^2)$ time to compute a bulge loop element, and worst case $\mathcal{O}(m^2K^6)$ time to determine the contribution of an internal loop element. To remedy the worst case $\mathcal{O}(m^2K^6)$ time, we follow other secondary structure prediction methods and employ a parameter L to bound the maximum interior loop size, including bulge loop and internal loop [9,12]. Then, the time to determine the contribution of an internal loop element can be reduced to $\mathcal{O}(mLK^6)$. Since there are $\mathcal{O}(m^2K^2)$ entries to compute, the overall time complexity of solving the dynamic program is $\mathcal{O}(m^2K^2) \cdot \mathcal{O}(mLK^6) = \mathcal{O}(m^3LK^8)$.

When disregarding CAI, i.e. $\lambda = 1$, we can adapt the recurrences such that for each entry $E[a][b][s][t][\mathbf{x}][\mathbf{y}]$, it would take $\mathcal{O}(1)$ time to compute a a stacking or a hairpin loop element, $\mathcal{O}(m)$ time to compute a bulge loop element, and worst case $\mathcal{O}(m^2)$ time to determine the contribution of an internal loop element. With a similar implementation of a maximum interior loop size L, the time to compute an internal loop element can be reduced to $\mathcal{O}(mL)$. Thus, the overall time complexity drops to $\mathcal{O}(m^2K^2) \cdot \mathcal{O}(mL) = \mathcal{O}(m^3LK^2)$ when $\lambda = 1$.

3.2 Pareto Optimal RNA Design

In the Pareto Optimal RNA Design problem (Problem 3), we are given a protein sequence $\mathbf{w} \in \Sigma_{prot}^m$ and seek a set of Pareto optimal solutions $(\mathbf{v}, P)$. We use the weighted sum method [30]. In this method, distinct convex combinations of the multiple objective functions is optimized. In our case this corresponds to solving distinct convex combinations of the two objectives MFE (Definition 9) and $\overline{\text{CAI}}$ (Equation (2)), which correspond to solving distinct instances of the Balanced RNA Design problem with varying values of the parameter $\lambda \in [0, 1]$. The weighted sum method has several limitations: (i) multiple λs may generate the same solution, (ii) the non-convex part of the Pareto front cannot be recovered, and (iii) there are non-uniform sampling issues [2,5].

We mitigate the first limitation by recursively examining λ values. More specifically, we maintain a queue Q of intervals $[\lambda^-, \lambda^+]$ as well as a hash table X such that $X[\lambda]$ yields the solution $(\mathbf{v}, P)$ of the Balanced RNA Design (BRD) problem instance $(\mathbf{w}, \lambda)$. Initially, Q contains a single interval $[\epsilon, 1 - \epsilon]$ where ϵ is a small constant (the default value in our implementation is $\epsilon = 10^{-5}$). Additionally, we initialize $X[\epsilon]$ and $X[1 - \epsilon]$ with the solutions of BRD problem instances $(\mathbf{w}, \epsilon)$ and $(\mathbf{w}, 1 - \epsilon)$, respectively. As long as the queue Q is not empty, we obtain an interval $[\lambda^-, \lambda^+]$ from Q, and solve a new BRD instance $(\mathbf{w}, \lambda)$ where $\lambda = \lambda^- + (\lambda^+ - \lambda^-)/2$, yielding solution $(\mathbf{v}, P)$. If this solution differs from $X[\lambda^-]$ and $X[\lambda^+]$, we set $X[\lambda] = (\mathbf{v}, P)$ and add (λ^-, λ) and (λ, λ^+) to the queue Q if $\lambda - \lambda^- > \tau$. We use a default value of 10^{-3} for the threshold parameter τ.

3.3 Implementation Details of DERNA

We implemented our algorithms for solving the BRD and PORD problems in C++11. The resulting method, DERNA (short for DEsign RNA), is available at `https://github.com/elkebir-group/derna.git` under the BSD 3-clause license. Usage instructions and examples are also available on the GitHub site.

DERNA uses the same energy model [15] as CDSfold [23]. For codon usage data, DERNA use the *Homo sapiens* codon usage table published in the codon usage database [18]. In addition, DERNA accepts alternative energy models and codon usage data in CSV format.

To validate the correctness of our algorithm and its implementation, we split our recurrences into two separate components and utilized two separate tables to store the MFE and the CAI separately. Using the real data instances examined in Section 4, for each solution $(\mathbf{v}, P)$ identified by DERNA, we confirmed that the MFE predicted by DERNA matched the MFE calculated using the Zuker algorithm [32] when given DERNA's inferred RNA sequence $\mathbf{v}$. Additionally, we recomputed the CAI of DERNA's inferred RNA sequence $\mathbf{v}$ and confirmed that the resulting value matched the CAI inferred by DERNA.

4 Results

We compare DERNA to CDSfold [23] and LinearDesign [31] on 100 protein sequences from the UniProt database [3] (Section 4.1) as well as on a case study involving the SARS-CoV-2 spike protein (Section 4.2). While the LinearDesign paper [31] describes both an exact and heuristic algorithm, only the heuristic algorithm was publicly available. As such, we were only able to include the heuristic algorithm in our benchmarking. All experiments were performed on a laptop with an Apple M1 Max 10-core CPU and 64 GB of RAM.

4.1 Benchmarking on 100 UniProt Protein Sequences

We begin by performing experiments that prioritize MFE over CAI in Section 4.1.1. In Section 4.1.2, we focus on the Pareto Optimal RNA Design problem, seeking solutions that collectively capture the trade-off between MFE and CAI.

4.1.1 Prioritizing MFE

The goal of this section is to assess the ability of RNA design methods to prioritize MFE over CAI. We seek solutions that achieve the minimum MFE and, as a secondary criterion, achieve largest CAI – i.e. among the space of solutions that achieve minimum MFE, we prefer those solutions that have the largest CAI value. We benchmarked using the same 100 protein sequences used in the CDSfold paper [23], which come from the UniProt database [3] and have lengths ranging from 78 to 2828 amino acids (Figure 4a). By design, CDSfold does not take CAI into account. Our method DERNA as well as LinearDesign support balancing MFE and CAI. For DERNA, we set $\lambda = 1 - \epsilon = 1 - 10^{-5}$. We note that LinearDesign's objective function is slightly different than DERNA's, seeking an RNA sequence $\mathbf{v}$ and secondary structure P that minimize $\text{MFE}(\mathbf{v}, P) - \lambda_{\text{LD}} \cdot \log \overline{\text{CAI}}(\mathbf{v}, \mathbf{w})$ for a target protein sequence $\mathbf{w}$. To similarly prioritize MFE, we set $\lambda_{\text{LD}} = \epsilon = 10^{-5}$ for LinearDesign and ran it with default parameters.

With the exception of the longest sequence (Q9NR99) with 2828 amino acids, which LinearDesign failed to complete within 24 hours (after which we killed the process), all methods ran successfully on all sequences. Moreover, with the exception of protein sequence Q9HAE3 (with 211 amino acids), all methods achieved the same minimum MFE (Figure 4b.

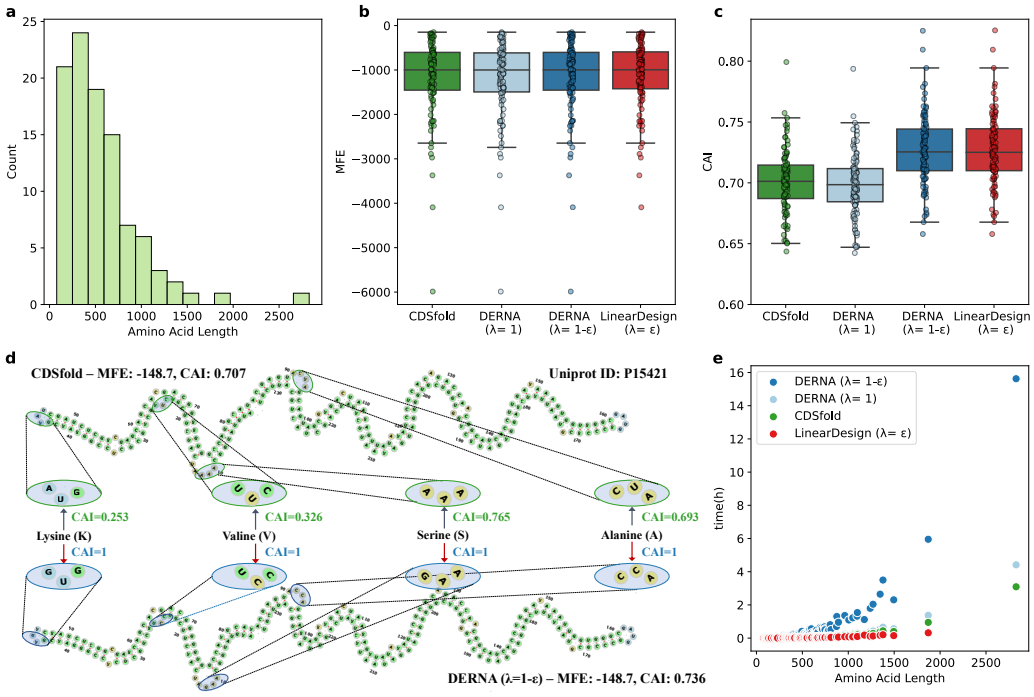

Figure 4 DERNA with $\lambda = 1 - \epsilon$ identifies solutions that achieve optimal MFE and largest CAI as a secondary objective. (a) We used 100 UniProt sequences, with varying lengths as shown. (b) With one exception (discussed in the text), all methods returned solutions with the same MFE. (c) However, the CAI values differed drastically between methods, with DERNA ($\lambda = 1 - \epsilon$) and LinearDesign outperforming CDSfold. (d) As an example, we show protein sequence P15421 for which CDSfold (top) and DERNA (bottom) inferred the same MFE and identical secondary structures. However, the solutions contain different codons resulting in different CAI values. (e) Wall-clock running times.

However, the CAI values varied between methods. In particular, DERNA with $\lambda = \epsilon$ and LinearDesign λ_{LD} achieved larger CAI values than CDSfold for all instances (Figure 4c and Figure S1). This makes sense because CDSfold only optimizes MFE but not CAI. The improved CAI values suggest that the sequences generated using our approach may exhibit higher *in vivo* translational efficiency without sacrificing mRNA half-life [16].

To further illustrate this point, we highlight the results for protein sequence P15421 with 78 amino acids. Both CDSfold and DERNA achieved the same MFE value of -148.7, yielding identical secondary structures (in terms of complementary base pairings) consisting of mostly stacking elements that achieve the lowest MFE. CDSfold, however, identified a different RNA sequence than DERNA resulting in a CAI of 0.707 whereas DERNA achieved a CAI of 0.736. The two RNA sequences differ at four codons encoding four distinct amino acids. For each such amino acid, DERNA used the codon that achieved the largest CAI value. For example, for the first codon encoding for the amino acid lysine (K), DERNA used the codon GUG with a relative usage frequency of 1 whereas CDSfold used the codon GUA with a smaller relative frequency of 0.253. The other three codons differed in a similar fashion. We note that LinearDesign identified the same RNA sequence as DERNA for this instance.

As for the CAI values inferred by LinearDesign, these largely match those inferred by DERNA (Figure 4c and Figure S1). The only exception is protein sequence Q9HAE3 where LinearDesign performed better in terms of CAI with a value of 0.754 vs. 0.748 for

DERNA. The solution inferred by DERNA, however, has a better MFE of -369.9 vs -369.4 for LinearDesign. Using a smaller $\lambda = 0.062509 < 1 - \epsilon$, DERNA was able to recover LinearDesign's solution. On the other hand, we were not able to identify a λ_{LD} value for which LinearDesign would identify DERNA's solution that achieved better MFE. A potential reason for this is that publicly-available version of LinearDesign is not an exact algorithm.

Finally, we consider the running times of CDSfold, LinearDesign and DERNA. Leaving out the largest instance (for which LinearDesign failed), we found that LinearDesign was the fastest algorithm with running times ranging from 1.80 to 1149.02 seconds, followed by CDSfold ranging from 1.86 to 3411.91 seconds and then DERNA with running times ranging from 16 to 21434 seconds. It is important to note that DERNA is an exact algorithm, while the publicly-available version of LinearDesign is a heuristic utilizing beam search. Indeed, as discussed above there was one instance were LinearDesign returned a suboptimal solution (in terms of the lexicographical objective of prioritizing MFE first followed by CAI).

We note that the difference in running times between DERNA and CDSfold because DERNA takes into consideration both MFE and CAI whereas CDSfold only considers MFE. As discussed in Section 3.1.1, leaving out CAI from the objective value reduces the asymptotic running time from $O(m3LK^8)$ to $O(m^3LK^2)$ where m is the protein sequence length and K and L are constants corresponding to the maximum number of codons per amino acid and the maximum interior loop length, respectively. Indeed, this is also reflected in wall-clock times when running an altered version of DERNA that only considers MFE, reducing the running times to between 2 and 4951 seconds, closely matching those of LinearDesign (Figure 4e). As expected, however, this comes at the expense of decreased CAI values for the inferred RNA sequences (Figure 4c and Figure S1).

4.1.2 Balancing MFE and CAI

We now assess DERNA's ability to identify Pareto optimal solutions. To that end, we ran DERNA in λ-sweep mode with a termination threshold value of $\tau = 0.001$. Note that the number of λ values explored by DERNA depends on both the value of τ as well as the input instance itself. Unlike our method, LinearDesign does not include an automated way of altering their λ_{LD} parameter. As such, we manually varied $\lambda_{\text{LD}} \in \{10^{-10}, 10^{-3}, 0.1, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 100\}$ – we did not set λ_{LD} to the same, instance-specific λ values examined by DERNA as the two parameters play different roles in the corresponding objective functions of both methods. Due to an increased number of runs per instance, we restricted our analysis to the 50 smallest instances with lengths ranging from 78 to 494 amino acids.

We begin by discussing the results for protein sequence P15421, which has 78 amino acids. DERNA examined 27 distinct λ values, leading to 12 distinct solutions (Figure 5a). On the other hand, the list of 14 λ_{LD} values resulted in 9 distinct solutions identified by LinearDesign. Recall that $\lambda = 1$ prioritizes MFE for DERNA whereas $\lambda = 0$ prioritizes CAI. Moreover, recall that each value of $\lambda \in [0, 1]$ leads to a Pareto optimal solution. A natural question is what is the smallest value λ^{MFE} that resulted in the optimal MFE? For protein sequence P15421 this was $\lambda^{\text{MFE}} = 0.0371186$. Given that $\tau = 0.001$, this means that DERNA does not explore the part of the Pareto front that contains solutions with higher CAI values. Indeed, for this protein sequence, the largest non-optimal CAI value identified by DERNA equals 0.968613, obtained using $\lambda = 0.000987$, followed by a CAI of 0.923779 using $\lambda = 0.001963$. On the other hand, the largest non-optimal CAI value identified by LinearDesign equals 0.991, which was obtained using $\lambda_{\text{LD}} = 10$, with a total of 7 solutions that have a CAI of at least 0.923779. A downside of LinearDesign's objective function, which

is of the form $\text{MFE}(\mathbf{v}, P) - \lambda_{\text{LD}} \cdot \log \overline{\text{CAI}}(\mathbf{v}, \mathbf{w})$, is that a non-bounded $\lambda_{\text{LD}} = \infty$ is required to exclusively prioritize CAI as opposed to a bounded value of $\lambda = 1$ for DERNA. Here, LinearDesign obtained this CAI-optimal solution only using $\lambda_{\text{LD}} = 100$.

We now extend these analyses to all 50 protein sequences. First, we observed that the median value of λ_{MFE} – the smallest λ that produces an MFE optimal solution – equals 0.0546964. Second, for $\tau = 0.001$, the median number of λs examined by DERNA is 36 (Figure S2a), yielding a median number of 17 solutions (Figure S2b). On the other hand, the 14 λ_{LD} examined by LinearDesign yielded a median number of 13 solutions (Figure S2c). To compare MFEs across instances, we define the MFE percentage as $(\text{MFE}(\mathbf{w}, \lambda) - \text{MFE}(\mathbf{w}, 0)/(\text{MFE}(\mathbf{w}, 1) - \text{MFE}(\mathbf{w}, 0))$ for each protein sequence $\mathbf{w}$ where $\text{MFE}(\mathbf{w}, \lambda)$ equals the MFE value of the solution obtained using λ. In other words, an MFE percentage of 100% means that the identified solution achieved the best possible MFE whereas an MFE percentage of 0% means that the worst MFE that favors CAI was obtained. We define CAI percentage similarly. Matching the previous analysis, we indeed see that DERNA favored the part of the Pareto front that prioritizes MFE (Figure 5b). Conversely, for our choices of λ_{LD}, LinearDesign more heavily favored the part of the Pareto front that prioritizes CAI (Figure 5c).

Finally, we delve more into the trade-off between CAI and MFE. To that end, we explored the following two questions. First, if one is willing to accept a certain CAI percentage, what is the best MFE that one can obtain? Second, for a specified minimum MFE percentage, what is the best CAI that one can obtain? Among the 50 considered instances, we found that if we accept solutions with a CAI percentage of at least 50% the corresponding best MFE percentages for these solutions identified by DERNA range from 81.298% to 92.65% with a median of 88.066% (Figure 5e). However, increasing the minimum CAI percentage to at least 80%, resulted in a decrease in best MFE of solutions identified by DERNA, with MFE percentages ranging from 55.624% to 72.965% with a median of 61.263%. Conversely, for an MFE percentage of at least 50%, DERNA obtained solutions that have CAI percentages ranging from 86.403% to 91.386% with a median of 87.874% (Figure 5f). Increasing the minimum MFE percentage to at least 80%, resulted in a decrease in best CAI of solutions identified by DERNA, with CAI percentages ranging from 54.443% to 71.362% with a median of 61.075%. When designing an RNA sequence for a target protein it is important to understand the trade-off between MFE and CAI, especially when trying to identify a single solution on the Pareto front.

4.2 Case Study: SARS-CoV-2 Spike Protein

The spike (S) protein on the surface of the SARS-CoV-2 virus is responsible for recognizing and binding to the host cell's receptors, as well as merging itself with the host cell membrane, without which the virus would be unable to interact with the host cells and initiate infection [10]. The SARS-CoV-2 S protein, with its 1273 amino acids, is therefore the primary target of the Moderna and Pfizer-BioNTech mRNA vaccines [20].

We applied LinearDesign to the S protein using a list of manually set values for λ_{LD}, specifically $\lambda_{\text{LD}} \in \{10^{-10}, 10^{-3}, 0.1, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 100\}$. LinearDesign generated 14 distinct solutions corresponding to the 14 chosen λ values. Similarly, we ran DERNA on the S protein with termination threshold $\tau = 0.0001$, ten times smaller than the previous analysis in Section 4.1.2. DERNA evaluated 76 distinct λ values and generated 56 distinct solutions. The set of solutions obtained through LinearDesign overlaps with those generated by DERNA (Figure 6a), with 3 identical solutions identified by both LinearDesign and DERNA.

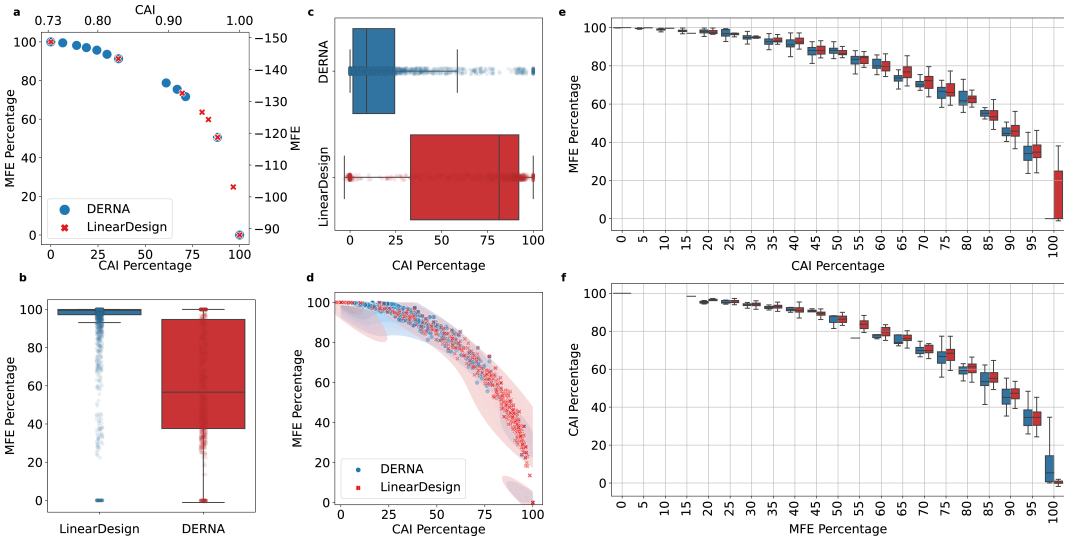

Figure 5 DERNA models the trade-off between MFE and CAI. These analyses are restricted to the 50 smallest UniProt sequences. (a) Solutions identified by DERNA (blue) and LinearDesign (red) for proteins sequence P15421. The right y-axis shows the MFE whereas the left y-axis shows the range-normalized MFE percentage. Similarly, the top x-axis shows the CAI wheres the bottom x-axis shows the range-normalized CAI percentage. (b-d) MFE and CAI percentages inferred by both methods across all 50 instances. (e) For each instance, we show the best MFE percentage on the y-axis when only considering solutions that achieve the CAI percentage specified on the x-axis. (f) For each instance, we show the best CAI percentage on the y-axis when only considering solutions that achieve the MFE percentage specified on the x-axis.

Finally, we compared DERNA's solutions to the Pfizer-BioNTech and Moderna mRNA sequences. The Pfizer-BioNTech mRNA sequence has an MFE of -1217 and a CAI of 0.95 (Figure 6b). For the same CAI value, DERNA identified a solution with a better MFE of -1955.2 (Figure 6c). On the other hand, the Moderna mRNA sequence has an MFE of -1369.2 and a CAI of 0.98. Similarly, for the same CAI value, DERNA identified a solution with a better MFE of -1724.8 These two alternative solutions might lead to increased mRNA half-life without sacrificing translational efficacy [16, 24]. We note that the overall minimum MFE equals -2486.7 with a corresponding CAI of 0.737 (Figure S3a), whereas solutions with overall maximum CAI of 1 lead to a decreased best MFE of -1384.3 (Figure S3b).

5 Discussion

Given a target protein sequence $\mathbf{w}$, we introduced the Pareto Optimal RNA Design (PORD) problem of identifying a set of Pareto optimal solutions $(\mathbf{v}, P)$ composed of an RNA sequence $\mathbf{v}$ that encodes for $\mathbf{w}$ and its corresponding secondary structure P that together balance the minimum free energy (MFE) and codon adaptation index (CAI). In addition, we introduced the Balanced RNA Design (BRD) problem, where we additionally take as input the parameter $\lambda \in [0, 1]$ and return an RNA sequence $\mathbf{v}$ whose corresponding secondary structure P minimizes $\lambda \cdot \text{MFE}(\mathbf{v}, P) - (1 - \lambda) \cdot \overline{\text{CAI}}(\mathbf{v}, \mathbf{w})$. To solve both problems, we introduced DERNA (DEsign RNA). Building on the work of Zuker and Stiegler [32], DERNA solves the BRD problem via dynamic programming in $\mathcal{O}(|\mathbf{w}|^3)$ time and $\mathcal{O}(|\mathbf{w}|^2)$ space. In addition, DERNA solves the PORD problem via the weighted sum method [30], enumerating the Pareto front by solving multiple distinct instances of the BRD problem via a systematic

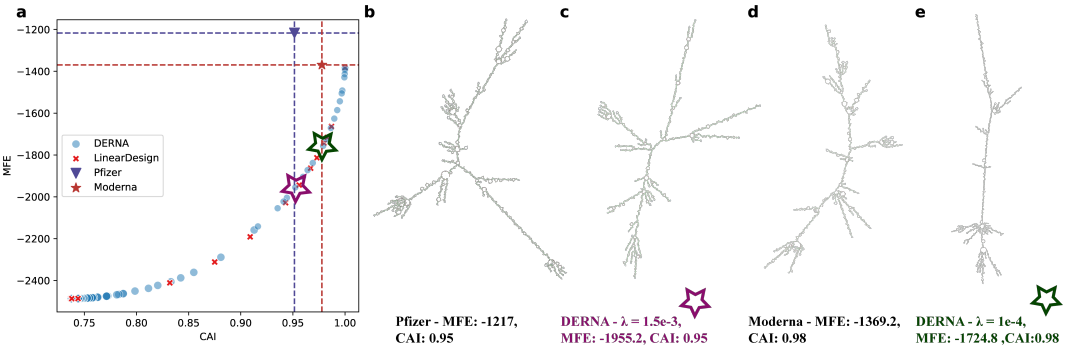

Figure 6 DERNA identifies alternative sequences for the SARS-CoV-2 spike (S) protein. (a) Solution identified by DERNA (blue) and LinearDesign (red). (b) Secondary structures of the Pfizer-BioNTech and Moderna mRNA vaccine sequences and alternative solutions provided by DERNA, from left to right are Pfizer-BioNTech, DERNA with $\lambda = 1.5 \cdot 10^{-3}$, Moderna, and DERNA with $\lambda = 10^{-4}$ respectively.

sweep on λ. On a benchmark dataset of 100 protein sequences, we demonstrated that DERNA obtained solutions with identical MFE but superior CAI compared to CDSfold [23], a previous approach that only optimizes MFE. Additionally, we showed that DERNA matched LinearDesign's performance in terms of solution quality, a recent approach that similarly seeks to balance MFE and CAI. While LinearDesign demonstrated better performance in terms of runtime, it is important to note that it employs a parameter-dependent algorithm that produces heuristic outcomes, whereas DERNA is guaranteed to solve the problem to optimality. In addition, key functionality of LinearDesign is closed source, whereas DERNA is fully open source. Finally, we demonstrated our method's potential for mRNA vaccine design using SARS-CoV-2 spike as the target protein.

For future development, it would be beneficial to integrate additional secondary structures beyond the five already considered in the algorithm, such as dangling ends. In particular, dangling ends allow one to capture the importance of 5' end in mRNA stability. That is, several studies have shown that secondary structure near the 5' untranslated region leads to decreased translation initiation and therefore decreased translational efficiency [6, 25, 27]. It will be particularly interesting to identify RNA sequences whose best MFE secondary structure lacks secondary structure at the 5' – this will probably require similar techniques as employed in traditional RNA design where one seeks an RNA sequence that folds into a desired RNA secondary structure [9, 11]. Finally, it will be valuable to investigate computing the Pareto front through algebraic dynamic programming [21].

References

1 Barry Cohen and Steven Skiena. Natural selection and algorithmic design of mRNA. *Journal of Computational Biology*, 10(3-4):419–432, 2003.

2 Jared L Cohon. *Multiobjective programming and planning*, volume 140. Courier Corporation, 2004.

3 The UniProt Consortium. UniProt: the Universal Protein Knowledgebase in 2023. *Nucleic Acids Research*, 51(D1):D523–D531, November 2022. doi:10.1093/nar/gkac1052.

4 Daan JA Crommelin, Thomas J Anchordoquy, David B Volkin, Wim Jiskoot, and Enrico Mastrobattista. Addressing the cold reality of mRNA vaccine stability. *Journal of Pharmaceutical Sciences*, 110(3):997–1001, 2021. doi:10.1016/j.xphs.2020.12.006.

5 I. Das and J. E. Dennis. A closer look at drawbacks of minimizing weighted sums of objectives for Pareto set generation in multicriteria optimization problems. *Structural Optimization*, 14(1):63–69, August 1997. doi:10.1007/BF01197559.

6 Yiliang Ding, Yin Tang, Chun Kit Kwok, Yu Zhang, Philip C Bevilacqua, and Sarah M Assmann. In vivo genome-wide profiling of RNA secondary structure reveals novel regulatory features. *Nature*, 505(7485):696–700, 2014.

7 Susan M Freier, Ryszard Kierzek, John A Jaeger, Naoki Sugimoto, Marvin H Caruthers, Thomas Neilson, and Douglas H Turner. Improved free-energy parameters for predictions of RNA duplex stability. *Proceedings of the National Academy of Sciences*, 83(24):9373–9377, 1986. doi:10.1073/pnas.83.24.9373.

8 Claes Gustafsson, Sridhar Govindarajan, and Jeremy Minshull. Codon bias and heterologous protein expression. *Trends in Biotechnology*, 22(7):346–353, 2004. doi:10.1016/j.tibtech.2004.04.006.

9 Ivo L Hofacker, Walter Fontana, Peter F Stadler, L Sebastian Bonhoeffer, Manfred Tacker, Peter Schuster, et al. Fast folding and comparison of RNA secondary structures. *Monatshefte fur chemie*, 125:167–167, 1994.

10 Yuan Huang, Chan Yang, Xin-feng Xu, Wei Xu, and Shu-wen Liu. Structural and functional properties of sars-cov-2 spike protein: potential antivirus drug development for covid-19. *Acta Pharmacologica Sinica*, 41(9):1141–1149, 2020.

11 Robert Kleinkauf, Martin Mann, and Rolf Backofen. antaRNA: ant colony-based RNA sequence design. *Bioinformatics*, 31(19):3114–3121, 2015.

12 Rune B Lyngso, Michael Zuker, and CN Pedersen. Fast evaluation of internal loops in RNA secondary structure prediction. *Bioinformatics (Oxford, England)*, 15(6):440–445, 1999. doi:10.1093/bioinformatics/15.6.440.

13 Elisabeth Mahase. Covid-19: Moderna vaccine is nearly 95% effective, trial involving high risk and elderly people shows. *BMJ: British Medical Journal (Online)*, 371, 2020.

14 David H Mathews, Matthew D Disney, Jessica L Childs, Susan J Schroeder, Michael Zuker, and Douglas H Turner. Incorporating chemical modification constraints into a dynamic programming algorithm for prediction of RNA secondary structure. *Proceedings of the National Academy of Sciences*, 101(19):7287–7292, 2004. doi:10.1073/pnas.0401799101.

15 David H Mathews, Jeffrey Sabina, Michael Zuker, and Douglas H Turner. Expanded sequence dependence of thermodynamic parameters improves prediction of rna secondary structure. *Journal of molecular biology*, 288(5):911–940, 1999. doi:10.1006/jmbi.1999.2700.

16 David M Mauger, B Joseph Cabral, Vladimir Presnyak, Stephen V Su, David W Reid, Brooke Goodman, Kristian Link, Nikhil Khatwani, John Reynders, Melissa J Moore, et al. mRNA structure regulates protein expression through changes in functional half-life. *Proceedings of the National Academy of Sciences*, 116(48):24075–24083, 2019. doi:10.1073/pnas.1908052116.

17 SA Meo, IA Bukhari, J Akram, AS Meo, and D COVID Klonoff. COVID-19 vaccines: comparison of biological, pharmacological characteristics and adverse effects of pfizer/biontech and moderna vaccines. *Eur Rev Med Pharmacol Sci*, pages 1663–1669, 2021.

18 Yasukazu Nakamura, Takashi Gojobori, and Toshimichi Ikemura. Codon usage tabulated from international dna sequence databases: status for the year 2000. *Nucleic acids research*, 28(1):292–292, 2000. doi:10.1093/nar/28.1.292.

19 Vladimir Presnyak, Najwa Alhusaini, Ying-Hsin Chen, Sophie Martin, Nathan Morris, Nicholas Kline, Sara Olson, David Weinberg, Kristian E. Baker, Brenton R. Graveley, and Jeff Coller. Codon optimality is a major determinant of mRNA stability. *Cell*, 160(6):1111–1124, 2015. doi:10.1016/j.cell.2015.02.029.

20 Giovanni Salvatori, Laura Luberto, Mariano Maffei, Luigi Aurisicchio, Giuseppe Roscilli, Fabio Palombo, and Emanuele Marra. Sars-cov-2 spike protein: an optimal immunological target for vaccines. *Journal of translational medicine*, 18(1):222, 2020. doi:10.1186/s12967-020-02392-y.

21 Cédric Saule and Robert Giegerich. Pareto optimization in algebraic dynamic programming. *Algorithms for Molecular Biology*, 10(1):1–20, 2015.

22 Paul M Sharp and Wen-Hsiung Li. The codon adaptation index-a measure of directional synonymous codon usage bias, and its potential applications. *Nucleic acids research*, 15(3):1281–1295, 1987. doi:10.1093/nar/15.3.1281.

23 Goro Terai, Satoshi Kamegai, and Kiyoshi Asai. CDSfold: an algorithm for designing a protein-coding sequence with the most stable secondary structure. *Bioinformatics*, 32(6):828–834, 2016. doi:10.1093/bioinformatics/btv678.

24 Tamir Tuller, Yedael Y. Waldman, Martin Kupiec, and Eytan Ruppin. Translation efficiency is determined by both codon bias and folding energy. *Proceedings of the National Academy of Sciences*, 107(8):3645–3650, 2010. doi:10.1073/pnas.0909910107.

25 Tamir Tuller and Hadas Zur. Multiple roles of the coding sequence 5' end in gene expression regulation. *Nucleic acids research*, 43(1):13–28, 2015.

26 Douglas H Turner, Naoki Sugimoto, and Susan M Freier. RNA structure prediction. *Annual review of biophysics and biophysical chemistry*, 17(1):167–192, 1988.

27 Yue Wan, Kun Qu, Qiangfeng Cliff Zhang, Ryan A Flynn, Ohad Manor, Zhengqing Ouyang, Jiajing Zhang, Robert C Spitale, Michael P Snyder, Eran Segal, et al. Landscape and variation of RNA secondary structure across the human transcriptome. *Nature*, 505(7485):706–709, 2014.

28 Hannah K Wayment-Steele, Do Soon Kim, Christian A Choe, John J Nicol, Roger Wellington-Oguri, Andrew M Watkins, R Andres Parra Sperberg, Po-Ssu Huang, Eterna Participants, and Rhiju Das. Theoretical basis for stabilizing messenger RNA through secondary structure design. *Nucleic Acids Research*, 49(18):10604–10617, September 2021. doi:10.1093/nar/gkab764.

29 David E. Weinberg, Premal Shah, Stephen W. Eichhorn, Jeffrey A. Hussmann, Joshua B. Plotkin, and David P. Bartel. Improved ribosome-footprint and mrna measurements provide insights into dynamics and regulation of yeast translation. *Cell Reports*, 14(7):1787–1799, 2016. doi:10.1016/j.celrep.2016.01.043.

30 L. Zadeh. Optimality and non-scalar-valued performance criteria. *IEEE Transactions on Automatic Control*, 8(1):59–60, 1963. doi:10.1109/TAC.1963.1105511.

31 He Zhang, Liang Zhang, Ang Lin, Congcong Xu, Ziyu Li, Kaibo Liu, Boxiang Liu, Xiaopin Ma, Fanfan Zhao, Huiling Jiang, et al. Algorithm for optimized mRNA design improves stability and immunogenicity. *Nature*, pages 1–3, 2023. doi:10.1038/s41586-023-06127-z.

32 Michael Zuker and Patrick Stiegler. Optimal computer folding of large RNA sequences using thermodynamics and auxiliary information. *Nucleic acids research*, 9(1):133–148, 1981. doi:10.1093/nar/9.1.133.

A Supplementary Methods

A.1 Turner Energy Function

In this section, we give detailed definitions for $f_{\mathrm{s}}, f_{\mathrm{h}}, f_{\mathrm{b}}, f_{\mathrm{i}}$ and f_{m} based on the Turner energy model [14]. Let $\mathbf{v} \in \Sigma_{\mathrm{rna}}^n$ be an RNA sequence. Recall that $\Gamma = \{(A,U),(U,A),(G,C),(C,G),(G,U),(U,G)\}$ is the set of allowed base pairings.

We begin with f_{s}, which takes in two base pairings $(v_i, v_j), (v_{i+1}, v_{j-1}) \in \Gamma$. Then, f_{s} computes the free energy contributed by the stacking element as

$$f_{\mathrm{s}}(v_i v_j, v_{i+1}, v_{j-1}) = \mathrm{stacking}[(v_i, v_j)][(v_{i+1}, v_{j-1})]$$

where $\mathrm{stacking} : \Gamma \times \Gamma \to \mathbb{R}$ is a lookup table with experimentally measured element energies.

For the hairpin element, f_{h} takes in the base pairing $(v_i, v_j) \in \Gamma$, the unpaired nucleotides v_{i+1}, v_{j-1}, and the length of the hairpin loop $l = j - i$. Then, f_{h} yields the free energy contributed by the hairpin loop as

$$f_{\mathrm{h}}(v_i, v_j, v_{i+1}, v_{j-1}, l) = \mathrm{hairpin}[l] + \mathrm{mismatchH}[(v_i, v_j])][v_{i+1}][v_{j-1}] \\ + \mathbb{1}\{l = 3 \wedge (v_i, v_j) \in \{(A,U),(U,A)\}\} \cdot D$$

where $\mathrm{hairpin} : \mathbb{N} \to \mathbb{N}$ and $\mathrm{mismatchH} : \Gamma \times \Sigma_{\mathrm{rna}} \times \Sigma_{\mathrm{rna}} \to \mathbb{R}$ are lookup tables with free energies for the length of the hairpin and paired and their directly adjacent nucleotides, respectively. Finally, D is an additional penalty term applied to AU base pairings.

For the bulge loop element, f_{b} takes in two base pairings $(v_i, v_j), (v_{p_1}, v_{q_1}) \in \Gamma$ and the length of the bulge loop $l = \max(j - i, q_1 - p_1)$. The free energy f_{b} contributed by the bulge loop equals

$$f_{\mathrm{b}}(v_i, v_j, v_{p_1}, v_{q_1}, l) = \mathrm{bulge}[l] + \mathbb{1}\{(v_i, v_j) \in \{(A,U),(U,A)\}\} \cdot D \\ + \mathbb{1}\{(v_{p_1}, v_{q_1}) \in \{(A,U),(U,A)\}\} \cdot D$$

where $\mathrm{bulge} : \mathbb{N} \to \mathbb{R}$ is a lookup table with free energies for the length of the hairpin, and D is an additional penalty term applied to AU base pairings.

For the internal loop element, f_{i} takes in takes in two base pairings $(v_i, v_j), (v_{p_1}, v_{q_1}) \in \Gamma$, the unpaired nucleotides $v_{i+1}, v_{j-1}, v_{p_1-1}, v_{q_1+1}$, and the length of the left loop $ll = j - i$ as well as the length of the right loop $lr = q_1 - p_1$. The free energy $f_{\mathrm{i}}(v_i, v_j, v_{p_1}, v_{q_1}, v_{i+1}, v_{j-1}, v_{p_1-1}, v_{q_1+1}, ll, lr)$ contributed by the internal loop equals

$$\mathrm{mismatchI}[(v_i, v_j)][(v_{p_1}, v_{q_1})][v_{i+1}][v_{j-1}][v_{p_1-1}][v_{q_1+1}] + \mathrm{internal}[ll + lr] + |ll - lr| \cdot E$$

where internal and mismatchI are lookup tables with experimentally measured energies, and E is a penalty applied to imbalanced loops. Note that the above equation is a simplification – in the actual implementation the used lookup table mismatchI may vary based on ll and lr.

A.2 Recurrences for Structural Elements

Due to space constraints we omit the precise definitions of the recurrences of the various structural elements.

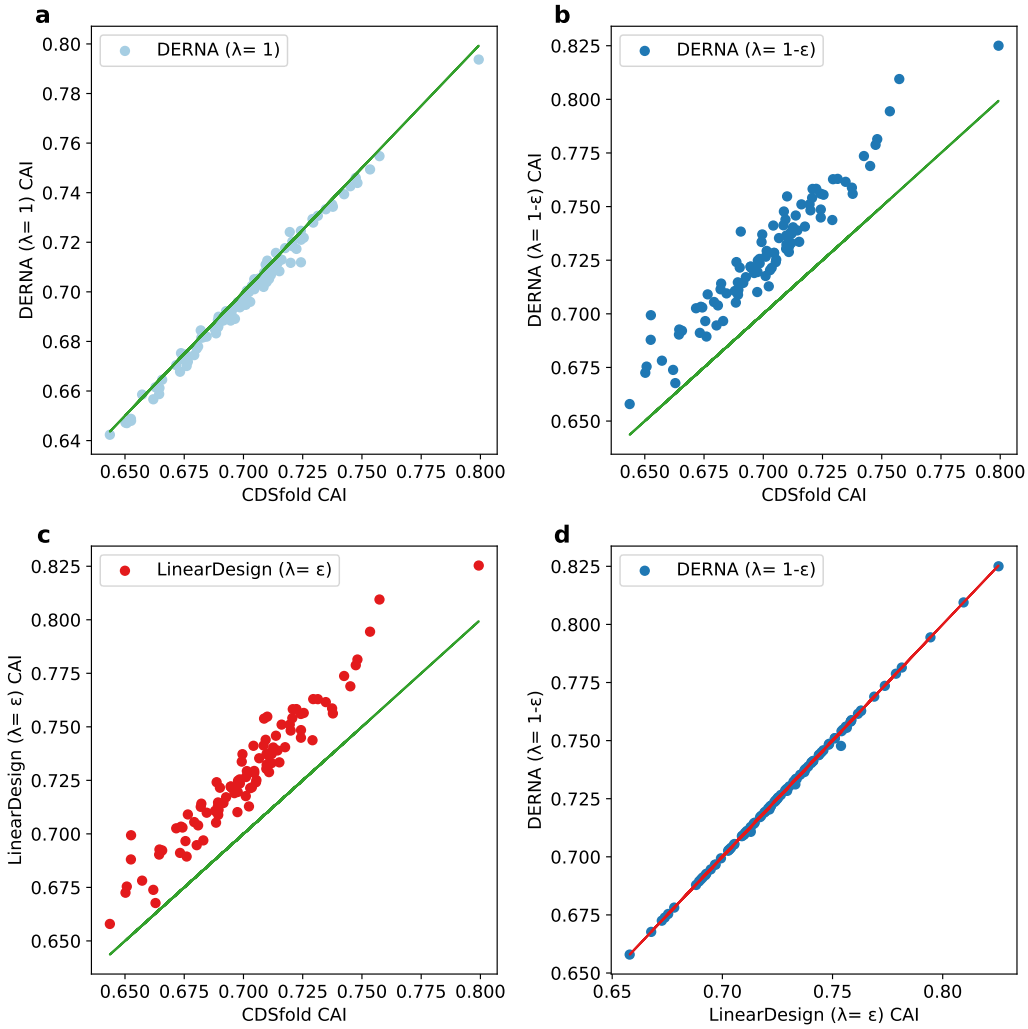

Figure S1 Comparison of CAI values for the benchmark dataset of 100 protein sequences. We compare CDSfold, DERNA with $\lambda = 1$, DERNA with $\lambda = 1 - \epsilon$ and LinearDesign with $\lambda_{\text{LD}} = \epsilon$. (a) DERNA with $\lambda = 1$ performs slightly worse than CDSfold in terms of CAI. However, neither method optimizes for CAI. (b-c) DERNA with $\lambda = 1 - \epsilon$ and LinearDesign achieve better CAI values than CDSfold. (d) With the exception of one protein sequence (Q9HAE3), discussed in the main text, LinearDesign and DERNA achieve the same CAI value. For protein sequence Q9HAE3, LinearDesign achieves a better CAI of 0.754 vs 0.748 for DERNA, but this comes at the expense of MFE (LinearDesign: −369.4 vs. DERNA: −369.9).

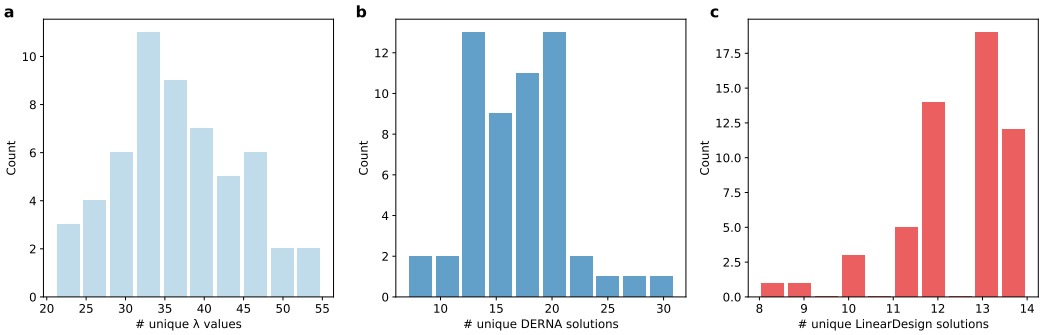

Figure S2 Distribution of (a) the number of unique λ values, (b) the number of unique solutions by DERNA, and (c) the number of unique solutions by LinearDesign for the dataset of 50 protein sequences.

Figure S3 DERNA identifies distinct mRNA sequences for SARS-CoV-2 S protein for (a) $\lambda = \epsilon$ (b) $\lambda = 1 - \epsilon$.