nature

https://doi.org/10.1038/s41586-023-06127-z

Accelerated Article Preview

Algorithm for Optimized mRNA Design Improves Stability and Immunogenicity

Received: 12 March 2022

Accepted: 25 April 2023

Accelerated Article Preview

Cite this article as: Zhang, H. et al. Algorithm for Optimized mRNA Design Improves Stability and Immunogenicity. *Nature* https://doi.org/10.1038/s41586-023-06127-z (2023)

He Zhang, Liang Zhang, Ang Lin, Congcong Xu, Ziyu Li, Kaibo Liu, Boxiang Liu, Xiaopin Ma, Fanfan Zhao, Huiling Jiang, Chunxiu Chen, Haifa Shen, Hangwen Li, David H. Mathews, Yujian Zhang & Liang Huang

Algorithm for Optimized mRNA Design Improves Stability and Immunogenicity

He Zhang[1,2†], Liang Zhang[1,2†], Ang Lin[3,8†], Congcong Xu[3†], Ziyu Li[1], Kaibo Liu[1,2], Boxiang Liu[1,9], Xiaopin Ma[3], Fanfan Zhao[3], Huiling Jiang[3], Chunxiu Chen[3], Haifa Shen[3], Hangwen Li[3*], David H. Mathews[4,5,6,7*], Yujian Zhang[3,10*], Liang Huang[1,2,7†*°]

[1]Baidu Research USA, Sunnyvale, CA 94089, USA, [2]School of EECS, Oregon State University, Corvallis, OR 97330, USA, [3]StemiRNA Therapeutics Inc., Shanghai 201206, China, [4]Dept. of Biochemistry and Biophysics, [5]Center for RNA Biology, and [6]Dept. of Biostatistics and Computational Biology, University of Rochester Medical Center, Rochester, NY 14642, USA, [7]Coderna.ai, Sunnyvale, CA 94085, USA, [8]Present address: Vaccine Center, School of Basic Medicine and Clinical Pharmacy, China Pharmaceutical University, Nanjing 211198, China, [9]Present address: Dept. of Pharmacy, National University of Singapore, Singapore 117543, [10]Present address: 660 Quince Orchard Rd. #1086, Gaithersburg, MD 20878, USA. †Equal contribution. *To whom correspondence should be addressed (°Lead contact)

E-mails: liang.huang.sh@gmail.com; yujianzhang@yahoo.com; david_mathews@urmc.rochester.edu; lihangwen@stemirna.com.

Messenger RNA (mRNA) vaccines are being used to contain COVID-19 (1, 2, 3), but still suffer from the critical limitation of mRNA instability and degradation, which is a major obstacle in the storage, distribution, and efficacy of the vaccine products (4). Previous work showed that increasing secondary structure lengthens mRNA half-life, which, together with optimal codons, improves protein expression (5). Therefore, a principled mRNA design algorithm must optimize both structural stability and codon usage. However, due to synonymous codons, the mRNA design space is prohibitively large (e.g., ~10^{632} candidates for the SARS-CoV-2 Spike protein), which poses insurmountable computational challenges. Here we provide a simple and unexpected solution using a classical concept in computational linguistics, where finding the optimal mRNA sequence is akin to identifying the most likely sentence among similar sounding alternatives (6). Our algorithm *LinearDesign* takes only 11 minutes for the Spike protein, and can jointly optimize stability and codon usage. On both COVID-19 and varicella-zoster virus mRNA vaccines, LinearDesign substantially improves mRNA half-life and protein expression, and dramatically increases antibody titer by up to 128× *in vivo*, compared to the codon-optimization benchmark. This surprising result reveals the great potential of principled mRNA design, and enables the exploration of previously unreachable but highly stable and efficient designs. Our work is a timely tool not only for

vaccines but also for mRNA medicine encoding all therapeutic proteins (e.g., monoclonal antibodies and anti-cancer drugs (7, 8)).

Messenger RNA (mRNA) vaccines (9, 10) have been recognized as viable tools to contain COVID-19 thanks to their scalable production, safety, and efficacy (1,2,3). However, an mRNA molecule is chemically unstable and prone to degrade, which leads to insufficient protein expression (5), and in turn, compromised immunogenicity and druggability. This instability has also become a major obstacle in the storage and distribution of the vaccine, requiring the use of cold-chain technologies that hinder its use in developing countries (4). Thus an mRNA molecule with enhanced stability is desired, which potentially has greater potency and favorable clinical efficacy.

While chemical stability is hard to model, previous work established its correlation with secondary structure, as quantified by the well-studied thermodynamic folding stability; improving this structural stability, combined with optimal codon usage, leads to increased protein expression (5). Therefore, a principled mRNA design algorithm must optimize two factors, structural stability and codon usage, to enhance protein expression.

However, the mRNA design problem (only considering the coding region in this work) is extremely challenging due to the exponentially large search space. Each amino acid is encoded by a triplet codon, i.e., three adjacent nucleotides, but due to redundancies in the genetic code (4^3 = 64 codons for 20 amino acids), most amino acids have multiple codons. This combinatorial explosion results in a prohibitively large number of candidates. For example, the Spike protein of SARS-CoV-2 with 1,273 amino acids can be encoded by ~2.4×10^{632} mRNA sequences (Fig. 1a). This poses an insurmountable computational challenge, and rules out enumeration which would take 10^{616} *billion years* for the Spike protein (Fig. 1b). On the other hand, the conventional approach to mRNA design, codon optimization (11, 12), only optimizes codon usage but barely improves stability, leaving out the huge space of highly stable mRNAs. Optimizing GC-content has a similar effect as it correlates with codon usage in vertebrates (13). As a result, the vast majority of highly stable designs remains unexplored.

Here we provide a simple algorithm, LinearDesign, to solve this challenging problem by an unexpected reduction to the classical concept of "lattice parsing" (6) in computational linguistics (Fig. 1c). We show that finding the optimal mRNA among the vast space of candidates is analogous to finding the most likely sentence among numerous similar-sounding alternatives. More specifically, we formulate the mRNA design space using a deterministic finite-state automaton (DFA), similar to a "word lattice" (6), which compactly encodes exponentially many mRNA candidates. We then use lattice parsing to find the most stable mRNA in the DFA, or the optimal balance between stability and codon optimality in a weighted DFA. This connection to natural language enables an efficient algorithm that scales quadratically with the mRNA sequence

length in practice. In this sense, our work turns the enormous search space into a blessing (freedom of design) rather than an obstacle.

Compared to the codon-optimized benchmark, both our COVID-19 and varicella-zoster virus (VZV) mRNA vaccines substantially improve chemical stability *in vitro*, protein expression *in cell*, and immunogenicity *in vivo*. In particular, our COVID-19 vaccines achieved up to 128× antibody response over the benchmark. This surprising result reveals the great potential of principled mRNA design, and enables the exploration of these previously unreachable but highly stable and efficient designs. Our work provides a timely and promising tool not only for mRNA vaccines, but also for mRNA therapeutics which has shown great potential to revolutionize healthcare (14), because LinearDesign can optimize mRNAs encoding all therapeutic proteins including monoclonal antibodies (7) and anti-cancer drugs (8).

Formulations and Algorithms

Previous work (5) established two main objectives for mRNA design, stability and codon optimality, which synergize to increase protein expression. To optimize for stability, given a protein sequence, we aim to find the mRNA sequence that has the *lowest minimum free energy change* (MFE) among all possible mRNA sequences encoding that protein, i.e., for each candidate mRNA sequence, we find its MFE structure among all its possible secondary structures using the standard RNA folding energy model (15, 16), and then choose the sequence whose MFE energy is the lowest. Therefore it is a *minimization within a minimization* (Extended Data Fig. 1a). But this naïve method would take billions of years, so we need an efficient algorithm without enumeration.

Next, we also aim to jointly optimize mRNA stability and codon optimality. The latter is often measured by the Codon Adaptation Index (CAI) (17) defined as the geometric mean of the relative adaptiveness of each codon in the mRNA. Because CAI is between 0 and 1 but MFE is generally proportional to the sequence length, we multiply the logarithm of CAI by the number of codons in the mRNA, and use a hyperparameter λ to balance MFE and CAI ($\lambda = 0$ being MFE-only). The combined objective is MFE $- \lambda |\mathbf{p}| \log \mathrm{CAI}$, where $|\mathbf{p}|$ is the protein length. See Methods §1.1 and Extended Data Fig. 1b for details.

We next describe our solution to these two optimization problems with two ideas borrowed from natural language: DFA (lattice) representation and lattice parsing.

Design Space Representation: DFA (Lattice) Inspired by the "word lattice" representation of ambiguities in computational linguistics (Extended Data Fig. 2a), we represent the choice of codons for each amino acid using a similar lattice, or more formally, a DFA, which is a directed graph with nucleotide-labeled edges (Fig. 2a & Extended Data Fig. 1c; see Methods §1.2 for formal definitions). After building a codon DFA for each amino acid in the protein sequence, we

concatenate them into a single mRNA DFA, where each path between the start and final states represents a possible mRNA sequence encoding that protein (Fig. 2b & Extended Data Fig. 1d).

Objective 1 (Stability): Lattice Parsing RNA folding is known to be equivalent to natural language parsing, where a stochastic context-free grammar (SCFG) can represent the folding energy model (18) (Extended Data Fig. 1e–f). But for mRNA design, the hard question is: how can all the mRNA sequences be folded in the DFA together? We borrow the idea of "lattice parsing" (19, 6), which generalizes single-sequence parsing to handle all sentences in the lattice simultaneously to find the most likely one (Fig. 1c & Extended Data Fig. 2). Similarly, we use lattice parsing to fold all sequences in the mRNA DFA simultaneously to find the most stable one (Fig. 2b & Extended Data Fig. 1g–h). Note that lattice parsing is also an instance of dynamic programming, but over a much larger search space, and single-sequence folding can be viewed as a special case of lattice parsing with a single-chain DFA. This process can also be interpreted as the SCFG-DFA intersection (Extended Data Fig. 1a) where the SCFG scores for stability and the DFA demarcates the set of candidates. This algorithm's runtime scales cubically with the mRNA sequence length (Methods §1.3), but for practical applications it only scales quadratically (Fig. 3a).

Adding Objective 2 (Codon Optimality): Lattice Parsing with Weighted DFAs We now extend DFAs to *weighted* DFAs (WDFAs) to integrate codon optimality on edge weights. Since our joint optimization formulation factors CAI onto the relative adaptiveness $w(c)$ of each individual codon c, we set edge weights in each codon DFA so that a codon c has path cost $-\log w(c)$, which can be interpreted as the "amount of deviation" from the optimal codon. Then in a weighted mRNA DFA, the cost of each start-end path is the sum of $-\log w(c)$ for each codon c in the corresponding mRNA, which is proportional to its –log CAI (Fig. 2d). Now lattice parsing takes a stochastic grammar (for stability) and a weighted DFA (for codon usage) and solves the joint optimization with optimality guarantee, which can be viewed as the weighted intersection (20) between an SCFG and a WDFA (Extended Data Fig. 1b; Methods §1.4).

Expressiveness of DFAs Our DFA framework is so general that it can also represent alternative genetic codes, modified nucleotides, and coding constraints. For details, see Methods §1.7, and Extended Data Fig. 3 and Supplementary Fig. 5.

Linear-time Approximation The exact design algorithm might still be slow for long sequences. Additionally, suboptimal designs may also be worth exploring for wet lab experiments, due to the many other factors involved in mRNA design besides stability and codon usage. So we developed an approximate search version that runs in linear time using beam search, keeping only the top b most promising items per step (b is the beam size), inspired by our previous work LinearFold (21).

Related Work Two previous studies also tackled the problem of "most stable mRNA design" (our objective 1) via dynamic programming, but using specialized extensions of the Zuker algorithm (22, 23) that cannot incorporate codon optimality (objective 2). By contrast, we established the connection between mRNA design and lattice parsing from computational

linguistics, which in our opinion is the most innovative contribution of our work. This connection enabled a simpler and more generalizable algorithm that can jointly optimize codon usage with a novel objective function that factors CAI onto individual codons. We also verified these algorithmic designs *in vivo*, showing substantial improvements for two mRNA vaccines (Figs. 4–5). See Methods §1.1 and §1.8 for details.

In silico Results and Analysis

Fig. 3a benchmarked the runtime of LinearDesign on UniProt proteins (24). LinearDesign was shown in a combination of two optimization objectives, MFE-only (objective 1) vs. MFE+CAI (objectives 1 & 2), and two search modes, exact search vs. beam search (beam size b=500). Empirically, LinearDesign scales quadratically with mRNA sequence length n for practical applications (n<10,000 *nt*) thanks to the DFA representation and lattice parsing (Supplementary Figs. 7–8). Next, our CAI-integrated exact search (CAI weight λ=4) had the same empirical complexity, and was only ~15% slower than the MFE-only version thanks to the convenience of our DFA representation for adding CAI. Last, our beam search version (b=500) further speeds up our design and scales linearly with sequence length, taking only 2.7 minutes (vs. 10.7 minutes for exact search) on the SARS-CoV-2 Spike protein (for MFE-only), with an approximation error (i.e., energy gap %, defined as $1–\mathrm{MFE}_{\mathrm{approx\_design}} / \mathrm{MFE}_{\mathrm{exact\_design}}$) of just 1.2%. In fact, as sequences get longer, this percentage stabilizes, suggesting that beam search quality does not degrade with sequence length (Supplementary Fig. 9).

For a GC-favoring codon preference (such as human), the conventional codon optimization method does improve stability, but only slightly (Figs. 3b–c), since its optimization direction (the pink arrows) are largely orthogonal to the stability optimization direction (the blue arrows). By contrast, our LinearDesign can directly optimize stability and find the optimally stable mRNAs. On both the COVID Spike protein and the VZV gE protein, the lowest MFEs (λ=0) are 1.8× lower than the optimal-CAI's (λ=∞). Also, our optimally stable designs have mostly double-stranded secondary structures (Fig. 3d), which are predicted to be much less prone to degradation (5). By varying λ from 0 to ∞, LinearDesign computes the feasibility limit (optimal boundary) of the mRNA design space (the blue curves in Fig. 3b–c; see Extended Data Fig. 4 for λ in (–∞, 0]). Furthermore, when the codon bias prefers AU-rich codons (such as in yeast), codon optimization actually worsens the stability (Extended Data Fig. 4b).

Results for COVID-19 mRNA Vaccines

For the COVID-19 Spike protein, eight mRNA sequences were employed in this study. Seven of them (sequences A-G) were designed with the LinearDesign algorithm as sub-optimal molecules (with beam search (21, 25)). They were widely distributed in the low-MFE design space (the region where MFE ≤ –1,400 kcal/mol in Fig. 4a) which is unreachable with conventional codon

optimizing algorithm. To have a better understanding of the biological impacts of MFE and CAI parameters, we designed these mRNA sequences to have almost identical values in either MFE (B–C // D–E–F) or CAI (A–C–F // B–E // D–G–H). The eighth mRNA sequence (sequence H) was designed with a widely-used codon optimization algorithm, OptimumGene™, as a benchmark. This benchmark sequence H has been used in a COVID-19 mRNA vaccine that showed high immunogenicity in two animal models (26) and entered a Phase I clinical trial in China (co-developed with China CDC; Chinese Clinical Trial Registry: CTR20210542). All mRNAs encode the same amino acid sequence of full-length SARS-CoV-2 wildtype Spike protein (S) and share the same 5'- and 3'-UTRs (see Supplementary Information for sequences).

Considering the potential negative impact on translation efficiency caused by a structured 5'-leader region (5), we did not include the first 5 amino acids when running LinearDesign, and instead used a heuristic to select the first 15 nucleotides. It is also suggested that long helices may elicit unwanted innate immune responses (27), so we avoided them in our designs. This also explains why we did not study the lowest-MFE candidates (closest to the optimal boundary – the blue curve in Fig. 4) which usually contain long stems. See Methods §1.10 for details.

Besides coding region design, UTR structure is also crucial for translation (28) and UTR engineering has a profound impact on protein expression (3). Although LinearDesign does not address UTR optimization per se, it does have an interesting property that its designed mRNA molecules, being more structured than codon-optimized ones, form fewer base pairs with, and thus interfere less with the structures of widely used UTRs (Extended Data Tab. 1). This speculation was confirmed by a different pair of UTRs in our VZV mRNA vaccine experiments (Extended Data Tab. 2), leading to improved protein expression and immune response (Fig. 5). These evidences suggest that LinearDesign is likely to remain effective independent of the choice of UTRs, which is also consistent with a recent study (29) where LinearDesign-generated sequences with three different UTRs all showed stronger *in vitro* protein expression over all benchmarks (see Fig. 4a of their paper); see Methods §1.8 for details.

In-solution Structure Compactness and Chemical Stability We then studied the structure compactness of mRNA molecules, which is hypothesized to be correlated with the folding free energy change. An mRNA molecule with a lower MFE tends to contain more secondary structures, exhibits more compact shape, and smaller hydrodynamic size. Therefore, it moves faster by electrophoresis. We loaded mRNA samples onto a non-denaturing agarose gel and found that RNA mobility rates correlated well with the calculated MFEs for sequences A–H (Fig. 4b) despite having similar molecular weights. Sequence A, with the lowest MFE, moved the fastest, followed by other sequences in order of their MFEs. Sequence H, with the highest MFE value, was the least mobile. This group of data demonstrated the validity of the MFE calculation executed by LinearDesign.

To evaluate chemical stability of mRNAs, we incubated the mRNAs in buffers containing 10 mM (Fig. 4c) or 20 mM (Extended Data Fig. 5) Mg^{2+} at 37 °C, and assessed RNA integrity following incubation, similar to previous work (29). Sequences A–H showed distinct degradation

rates that correlated well with their MFEs (Fig. 4c and Extended Data Fig. 5). Sequence A, with the lowest MFE, showed the slowest degradation rate with half-life ($T_{1/2}$) of 20.0 and 12.6 h in 10 and 20 mM Mg^{2+} buffers, respectively (Fig. 4c and Extended Data Fig. 5). By contrast, sequence H that has the highest MFE value degraded the fastest with $T_{1/2}$ of 3.9 and 3.3 h in 10 and 20 mM Mg^{2+} buffers, respectively. These results support that low-MFE designs are more resistant to in-solution degradation, rendering favorable biological significance.

Cellular Protein Expression For vaccine product, sufficient antigen expression is one of the key determinants for eliciting effective immune responses. We next evaluated protein expression of the designed mRNAs. All sequences A–H can be efficiently translated into S protein following transfection into HEK293 cells. Of note, all 7 LinearDesign-generated mRNAs (sequences A-G) showed remarkably higher protein expression levels than benchmark sequence H (Fig. 4d and Supplementary Fig. 12). Sequences D and G (with CAIs almost identical to H, but lower MFEs) expressed 2.3-fold more protein than sequence H, and sequence A with the lowest MFE showed 2.9-fold better expression. Collectively, our results are consistent with Mauger et al. (5) that low MFE and high CAI synergize to improve protein expression, but we were able to test this hypothesis using mRNA molecules with much lower MFEs than they could, thanks to LinearDesign's ability to explore the previously unreachable design space.

***In vivo* Immunogenicity** We further tested whether these designs could endow a superior immunogenicity *in vivo*. Sequences A–H mRNAs were delivered by a lipid-based formulation *in vivo* (30), and both humoral and cellular immune response were evaluated. For each mRNA sequence, C57BL/6 mice were intramuscularly immunized with two doses of vaccines at an interval of 2 weeks. Levels of anti-Spike IgG, neutralizing antibodies (NAbs), and Spike-specific interferon-γ (IFN-γ)-secreting T cells were assessed. All mRNA molecules from LinearDesign were able to elicit robust Ab responses. By contrast, sequence H mRNA showed very limited ability to induce Abs (Fig. 4e–f). Similar results were also observed on the antigen-specific T cell response, where a robust Th1-biased T cell response was induced only by LinearDesign mRNAs (Fig. 4g). Sequences A–D, which are closer to the optimal boundary (shaded in Fig. 4a), led to a 57~128× increase in anti-Spike IgG Ab titers and a 9~20× increase in neutralizing Ab titers over the benchmark sequence H.

Since BNT162b2 from Pfizer/BioNTech is the most widely adopted COVID-19 mRNA vaccine, we excerpted its mRNA sequence and made a side-by-side comparison. To perform a reasonable comparison, we constructed a new mRNA molecule BNT which uses the same 5'- and 3'-UTRs as sequences A–H and protein coding sequence of BNT162b2. The "2P" amino acid mutation (31) in BNT162b2 was converted back to the wildtype sequence as in A–H. Among 4 sequences (A, C, H, and BNT) in the study, sequences a and c showed a remarkably lower degradation rate and higher protein expression than BNT mRNA (Extended Data Fig. 6). Note that BNT and H have very similar MFEs and CAIs (Fig. 4a) and very similar half-lives. Moreover, sequences A and C were able to elicit higher levels of anti-Spike IgG and NAbs than the two benchmark sequences H and BNT (Extended Data Fig. 7). Collectively, these data lead us to speculate that LinearDesign-

optimized mRNA molecules are more stable *in vivo*, which leads to improved protein expression and enhanced immunogenicity.

Results for VZV mRNA Vaccines

To further evaluate the generalizability of LinearDesign, we also applied the algorithm to the design of varicella-zoster virus (VZV) mRNA vaccine. VZV vaccine is considered as an effective approach to reduce the risk of shingles (32). Using the same strategy as for Spike mRNA design (Fig. 4a), we generated five mRNA sequences encoding full-length VZV gE protein (gE A–E). They are widely distributed in the high-thermostability region that was previously unexplored (Fig. 5a). These sequences were benchmarked to the gE-Ther sequence designed with a widely-used codon-optimization tool GeneOptimizer (33) developed by ThermoFisher Scientific. All these mRNAs, including wildtype mRNA (gE-WT), shared the same encoded amino acid sequence and 5'/3' UTRs. See Supplementary Information for the sequences. In line with the Spike mRNA data (Fig. 4b), gE-A mRNA having the lowest MFE showed higher mobility in a non-denaturing gel (Fig. 5b) and remarkably slower degradation rates with $T_{1/2}$ of 66.5 h in 10 mM (Fig. 5c) and 50.7 h in 20 mM (Extended Data Fig. 8a) Mg^{2+} buffers, which indicated a high chemical stability correlated with the compactness of molecules. By contrast, gE-Ther showed a $T_{1/2}$ of 10.9 h in 10 mM and 5.9 h in 20 mM Mg^{2+} buffers. We also noticed that gE mRNA molecules have overall better stabilities than Spike mRNAs due to the shorter length (34). In addition, most of LinearDesign-generated molecules (gE B-E) outperformed gE-Ther and WT in protein expression 48 h (Fig. 5d) and 24 h (Extended Data Fig. 8b) after HEK293 cell transfection. But it is intriguing that best performing mRNA molecules are gE B–D, which outperformed both gE-A with the lowest CAI and gE-E with the lowest MFE. This finding, again, emphasizes the importance of jointly optimizing CAI and MFE. The very best molecules are those with both favorable CAI and MFE values, in the area we highlighted the "sweet spot" region by light blue shading in Fig. 5a. Lastly, the immunogenicity of VZV mRNA molecules was evaluated in C57BL/6 mice. LinearDesign mRNA molecules (gE-B, C, and E) induced significantly higher levels of anti-gE IgG than gE-Ther and WT (Fig. 5e).

Discussion

An effective mRNA design strategy is of utmost importance, especially for the development of mRNA vaccines that have shown great promise in fighting against the current and future pandemics. However, it is extremely challenging due to the prohibitively large search space. We presented a simple solution by reducing the mRNA design problem to the classical problem of lattice parsing in computational linguistics. This highly unexpected analogy, which is the most innovative part of this work, resulted in an efficient algorithm that takes only 11 minutes for the SARS-CoV-2 Spike protein. It can also jointly optimize stability and codon usage, which is important for mRNA design according to the literature and our VZV experiments. Our

interdisciplinary approach is another example among the recent fruitful exchanges between linguistics and biology (35, 36).

The mRNA sequences generated by LinearDesign were comprehensively characterized in this study and demonstrated superiority over the commonly used codon optimization benchmark using two viral antigens across three attributes critical for vaccine performance: chemical stability, protein translation and *in vivo* immunogenicity. In particular, our designs for the Spike protein showed up to 128-fold increase in binding antibody levels over the codon optimization benchmark, and our VZV mRNA designs, with a different UTR pair, also showed substantial improvements. These results suggested the robustness of LinearDesign in optimizing coding region independent of UTR pairs. In fact, these two directions (coding region design and UTR engineering (3)) are complementary and can be combined in future work. It is worth noting that our designed mRNAs did not use chemical modification which is widely believed to be critical to the recent success of mRNA vaccines (37,38,10,1,2), yet still showed high levels of stability, translation efficiency, and immunogenicity, with the additional advantage of lower manufacturing cost. On the other hand, our algorithm is complementary to chemical modification and can be easily adapted to (and enjoy with the benefits of) modified nucleotides once the corresponding energy model is available. Our work only considered stability and codon usage, but thanks to the generalizability of the lattice representation, it can also be adapted to optimize other parameters relevant to mRNA design when becoming available. By unleashing the previously inaccessible region of highly stable and efficient sequences, LinearDesign is a timely and promising tool for mRNA vaccine development which is of utmost importance to the current and future pandemics. But more importantly, it is also a general and principled method for molecule design in the field of mRNA therapeutics and can be used for all therapeutic proteins including monoclonal antibodies and anti-cancer drugs.

Online content

Any methods, additional references, Nature Research reporting summaries, source data, extended data, supplementary information, acknowledgements, peer review information; details of author contributions and competing interests; and statements of data and code availability are available at https://doi.org/10.1038/s41586-023-00000-0

1. Baden LR, et al. (2021) Efficacy and safety of the mRNA-1273 SARS-CoV-2 vaccine. New England Journal of Medicine 384(5):403–416.

2. Polack FP, et al. (2020) Safety and efficacy of the BNT162b2 mRNA Covid-19 vaccine. New England journal of medicine 383(27):2603–2615.

3. Gebre MS, et al. (2022) Optimization of non-coding regions for a non-modified mRNA COVID-19 vaccine. Nature 601(7893):410–414.

4. Crommelin DJ, Anchordoquy TJ, Volkin DB, Jiskoot W, Mastrobattista E (2021) Addressing the cold reality of mRNA vaccine stability. Journal of Pharmaceutical Sciences 110(3):997–1001.

5. Mauger DM, et al. (2019) mRNA structure regulates protein expression through changes in functional half-life. Proceedings of the National Academy of Sciences U.S.A. 116(48):24075–24083.

6. Hall KB (2005) Best-first Word-lattice Parsing: Techniques for integrated syntactic language modeling. PhD Thesis, Brown University.

7. Schlake T, et al. (2019) mRNA: a novel avenue to antibody therapy? Molecular Therapy 27(4):773–784.

8. Reinhard K, et al. (2020) An RNA vaccine drives expansion and efficacy of claudin-CAR-T cells against solid tumors. Science 367(6476):446–453.

9. Wolff JA, et al. (1990) Direct gene transfer into mouse muscle *in vivo*. Science 247(4949):1465–1468.

10. Pardi N, Hogan MJ, Porter FW, Weissman D (2018) mRNA vaccines—a new era in vaccinology. Nature Reviews Drug Discovery 17(4):261–279.

11. Mauro VP, Chappell SA (2014) A critical analysis of codon optimization in human therapeutics. Trends in molecular medicine 20(11):604-13.

12. Gustafsson C, Govindarajan S, Minshull J (2004) Codon bias and heterologous protein expression. Trends in Biotechnology 22(7):346–353.

13. Nabiyouni M, Prakash A, Fedorov A (2013) Vertebrate codon bias indicates a highly GC-rich ancestral genome. Gene 519(1):113–119.

14. Sahin U, Karikó K, Türeci Ö (2014) mRNA-based therapeutics—developing a new class of drugs. Nature reviews Drug discovery 13(10):759–780.

15. Mathews DH, Sabina J, Zuker M, Turner DH (1999) Expanded sequence dependence of thermodynamic parameters improves prediction of RNA secondary structure. Journal of Molecular Biology 288(5):911–940.

16. Mathews DH, et al. (2004) Incorporating chemical modification constraints into a dynamic programming algorithm for prediction of RNA secondary structure. Proceedings of the National Academy of Sciences U.S.A. 101(19):7287–7292.

17. Sharp PM, Li WH (1987) The codon adaptation index-a measure of directional synonymous codon usage bias, and its potential applications. Nucleic Acids Research 15(3):1281–1295.

18. Durbin R, Eddy SR, Krogh A, Mitchison G (1998) Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids. (Cambridge University Press, Cambridge, UK).

19. Bar-Hillel Y, Perles M, Shamir E (1961) On formal properties of simple phrase structure grammars. Zeitschrift für Phonetik, Sprachwissenschaft und Kommunikationsforschung 14(2):143–172.

20. Nederhof MJ, Satta G (2003) Probabilistic Parsing as Intersection. Proceedings of the Eighth International Conference on Parsing Technologies pp. 137–148.

21. Huang L, et al. (2019) LinearFold: linear-time approximate RNA folding by 5'-to-3' dynamic programming and beam search. Bioinformatics 35(14): i295–i304.

22. Cohen B, Skiena S (2003) Natural selection and algorithmic design of mRNA. Journal of Computational Biology 10(3-4):419–432.

23. Terai G, Kamegai S, Asai K (2016) CDSfold: an algorithm for designing a protein-coding sequence with the most stable secondary structure. Bioinformatics 32(6):828–834.

24. Consertium U (2005) UniProt: a hub for protein information. Nucleic Acids Research 42: D204–D12.

25. Huang L, Sagae K (2010) Dynamic programming for linear-time incremental parsing. Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics pp. 1077–1086.

26. Yang R, et al. (2021) A core-shell structured COVID-19 mRNA vaccine with favorable biodistribution pattern and promising immunity. Signal transduction and targeted therapy 6(1):1–10.

27. Liu L, et al. (2008) Structural basis of toll-like receptor 3 signaling with double-stranded RNA. Science 320(5874):379–381.

28. Mignone F, Gissi C, Liuni S, Pesole G (2002) Untranslated regions of mRNAs. Genome biology 3(3):1–10.

29. Leppek K, et al. (2022) Combinatorial optimization of mRNA structure, stability, and translation for RNA-based therapeutics. Nature communications 13(1):1–22.

30. Rana MM (2021) Polymer-based nano-therapies to combat COVID-19 related respiratory injury: progress, prospects, and challenges. Journal of Biomaterials Science, Polymer Edition 32(9):1219–1249.

31. Wrapp D, et al. (2020) Cryo-EM structure of the 2019-nCoV spike in the prefusion conformation. Science 367(6483):1260–1263.

32. Cunningham AL, et al. (2016) Efficacy of the herpes zoster subunit vaccine in adults 70 years of age or older. New England Journal of Medicine 375(11):1019–1032.

33. Raab D, Graf M, Notka F, Schödl T, Wagner R (2010) The GeneOptimizer Algorithm: using a sliding window approach to cope with the vast sequence space in multiparameter DNA sequence optimization. Systems and Synthetic Biology 4(3):215–225.

34.Wayment-Steele HK, et al. (2021) Theoretical basis for stabilizing messenger RNA through secondary structure design. Nucleic acids research 49(18):10604–10617.

35. Hie B, Zhong ED, Berger B, Bryson B (2021) Learning the language of viral evolution and escape. Science 371(6526):284–288.

36. Madani A, et al. (2023) Large language models generate functional protein sequences across diverse families. Nature Biotechnology pp. 1–8.

37. Karikó K, Buckstein M, Ni H, Weissman D (2005) Suppression of RNA recognition by Toll-like receptors: the impact of nucleoside modification and the evolutionary origin of RNA. Immunity 23(2):165–175.

38. Karikó K, et al. (2008) Incorporation of pseudouridine into mRNA yields superior nonimmunogenic vector with increased translational capacity and biological stability. Molecular therapy 16(11):1833–1840.

Main Text Figures

Fig. 1 | Overview of mRNA coding region design for two well-established objectives, *stability* and *codon optimality*, using SARS-CoV-2 Spike protein as an example. a, The combinatorial nature of mRNA design due to codon degeneracy (~10^{632} mRNA sequences for the Spike protein; taking ~10^{616} billion years to enumerate). The pink and blue paths represent the wildtype and the optimally stable (i.e., lowest energy) sequences, respectively. **b**, The vastly different secondary structures between these two sequences, with the former being mostly single-stranded (prone to degradation in red loop regions) while the latter mostly double-stranded. Our algorithm takes just 11 minutes for this optimization. **c**, An analogy between linguistics (left) and biology (right), where deterministic finite-state automaton (DFA) and lattice parsing from the former were adapted to solve mRNA design. An mRNA DFA (inspired by "word lattice") compactly encodes all mRNA candidates, which are folded simultaneously by lattice parsing to find the optimal mRNA (Fig. 2). **d**, 2D visualization of the mRNA design space, with stability on the *x*-axis and codon optimality on the *y*-axis. The standard mRNA design method *codon optimization* improves codon usage (the pink arrow) but is unable to explore the vast high-stability region (left of the dashed line), which is exemplified by the COVID-19 vaccine products of BioNTech-Pfizer (○), Moderna (☆), and CureVac (▷). LinearDesign, by contrast, jointly optimizes stability and codon optimality (the blue curve, with λ being the weight of the latter). By considering other factors, we select seven of our designs (four shown here) for COVID-19 vaccine experiments (Fig. 4), which show substantially enhanced half-life and protein expression, and up to 128 antibody responses over the codon-optimized baseline (H). Experiments on the varicella-zoster virus (VZV) mRNA vaccine (on a different antigen, and with different UTRs) show similar improvements (Fig. 5), confirming the generalizability of LinearDesign.

Fig. 2 | Illustration of the LinearDesign algorithm. a, Codon DFAs. **b**, An mRNA DFA (bottom) and lattice parsing on that DFA (top). In the DFA, the optimal mRNA sequence under a simplified energy model is shown in the thick blue path, together with its optimal structure shown in the dot-bracket format ("●": unpaired; "(" and ")": base pairs). In lattice parsing, the brown and black arcs also depict base pairs (two GC pairs and two AU pairs), while the round trapezoidal shadings depict the decomposition of the optimal structure. Among all mRNA sequences encoded in the DFA, lattice parsing finds the optimal sequence with its optimal structure, achieving the lowest free energy under this energy model where GC and AU pairs have -3 and -2 kcal/mol, respectively (Extended Data Fig. 1e). Note that here we use the simplified energy model for illustration, but our implementation uses the nearest neighbor energy model. **c**, Another illustration of the optimal sequence and secondary structure in **b**. **d**, Joint optimization between stability and codon optimality, by integrating the latter in weighted DFAs. Top: bar charts showing the codon frequencies of threonine and serine. The relative adaptiveness *w(c)* of a codon *c* is the ratio between the frequencies of *c* and its most frequent codon (shown in stripes). Bottom: a weighted mRNA DFA encodes each candidate's CAI in the total weight of its corresponding path by using –log

$w(c)$, the cost of choosing codon c, as edge weights (Methods §1.1). This weighted DFA can be plugged back into lattice parsing for joint optimization between stability and codon optimality.

Fig. 3 | Computational characteristics of LinearDesign algorithm. a, Runtime analysis of mRNA design for UniProt proteins (Supplementary Tab. 1). Overall, our exact search only scales quadratically with sequence length in practice (Supplementary Figs. 7–8), and our MFE+CAI mode (with CAI weight λ=4) is only ~15% slower than our MFE-only version. Moreover, beam search (b=500) significantly speeds up long sequence design, with minor search errors (Supplementary Fig. 9). **b–c**, 2D (MFE–CAI) visualizations of designs for the COVID Spike (**b**) and VZV gE proteins (**c**), respectively (both using human codon preference). The blue curves form the feasibility limit (optimal boundary), by varying from 0 to ∞ (see Extended Data Fig. 4 for λ in (–∞, 0]). GC% are shown in parentheses. The human genome prefers GC-rich codons, therefore codon optimization (the pink arrows in **b–c**) also improves stability, but only marginally, as the two optimization directions (codon vs. stability) are largely orthogonal. By contrast, Extended Data Fig. 4b shows that with an AU-rich codon preference (e.g., yeast), codon optimization decreases stability. **d**, Secondary structures of the mRNA designs for COVID-19 Spike protein and VZV gE protein. The optimal-CAI designs (top, λ=∞) are largely single-stranded (~60% paired), while the optimally stable designs (bottom, λ=0) are mostly double-stranded (~80% paired). We also show intermediate designs (center, λ=4) that balance between stability and CAI.

Fig. 4 | Experimental evaluation of LinearDesign-generated mRNA sequences encoding SARS-CoV-2 Spike protein. a, Summary of chemical stability, protein expression of our mRNA designs (A–G) and their immunogenicity in the induction of anti-Spike IgG compared to the codon-optimized baseline (H). **b**, Non-denaturing agarose gel characterization of mRNA showing the correlation of gel mobility with minimum free energy; for gel source data, see Supplementary Fig. 1a. **c**, Chemical stability of mRNAs upon incubation in buffer (Mg^{2+} = 10 mM) at 37 °C. Percentage of intact mRNA is shown. Data is from three independent experiments. **d**, Protein expression levels of mRNAs determined by flow cytometry 48 hours after transfection into HEK-293 cells. Mean fluorescence intensity (MFI) values derived from three independent experiments are shown. Kruskal–Wallis analysis of variance (ANOVA) with Dunn's multiple comparisons test to H group was performed for statistical analysis. **e–g**, C57BL/6 mice (n=6) were immunized *i.m.* with two doses of formulated mRNA at a 2-week interval. Endpoint titer of anti-Spike IgG (**e**). Levels of neutralizing Abs against wide-type SARS-CoV-2 (**f**). Frequencies of IFN-γ-secreting T cells measured by ELISpot (**g**). A two-tailed Mann-Whitney U test was used for statistical analysis. $*p < 0.05$, $**p < 0.01$, $***p < 0.001$. Data are presented as mean ± s.d. (**c**, **d**), geometric mean ± geometric s.d. (**e**, **f**) or mean ± s.e.m. (**g**). See Source Data for details. See also Extended Data Figs. 5–7, Supplementary Figs. 10 and 12, and Supplementary Tab. 2.

Fig. 5 | Experimental evaluation of LinearDesign-generated mRNAs encoding VZV gE protein. **a**, Summary of chemical stability, protein expression of mRNA designs, and immunogenicity in the induction of anti-gE IgG. The "sweet spot" region is highlighted by light blue shading. **b**, Non-denaturing agarose gel characterization of mRNA showing the correlation of gel mobility with minimum free energy; for gel source data, see Supplementary Fig. 1b. **c**, Chemical stability of mRNAs upon incubation in buffer (Mg^{2+} = 10 mM) at 37 °C. Percentage of intact mRNA is shown. Data is from three independent experiments. **d**, Protein expression of mRNAs following transfection into HEK293 cells for 48 hours determined by flow cytometry. MFI values derived from three independent experiments are shown. Kruskal–Wallis analysis of variance (ANOVA) with Dunn's multiple comparisons test to gE-Ther group was performed for statistical analysis. **e**, C57BL/6 mice (n=5) were immunized *i.m.* with two doses of mRNA vaccines at a 2-week interval. Endpoint titer of anti-gE IgG is shown. A two-tailed Mann-Whitney U test was used for statistical analysis. *$p < 0.05$, **$p < 0.01$, ***$p < 0.001$. Data are presented as mean ± s.d. (**c**, **d**) or geometric mean ± geometric s.d. (**e**). See Source Data for details. See also Extended Data Fig. 8, Supplementary Fig. 11 and Supplementary Tab. 3.

Methods

§1 Details of LinearDesign Algorithm

§1.1 Optimization Objectives There are two objectives in mRNA design: stability and codon optimality. The optimal-stability mRNA design problem can be formalized as follows. Given a protein sequence $\mathbf{p} = p_0 \ldots p_{|p|-1}$ where each p_i is an amino acid residue, we find the optimal mRNA sequence $\mathbf{r}^*(\mathbf{p})$ that has the *lowest minimum folding free energy change* (MFE) among all possible mRNA sequences encoding that protein:

$$\mathbf{r}^*(\mathbf{p}) = \text{argmin}_{r \in mRNA(\mathbf{p})} \text{MFE}(\mathbf{r}) \tag{1}$$

$$\text{MFE}(\mathbf{r}) = \min_{\mathbf{s} \in structures(\mathbf{r})} \Delta G^\circ(\mathbf{r}, \mathbf{s}) \tag{2}$$

where $mRNA(\mathbf{p}) = \{\mathbf{r} \mid protein(\mathbf{r}) = \mathbf{p}\}$ is the set of candidate mRNA sequences, $structures(\mathbf{r})$ is the set of all possible secondary structures for mRNA sequence $\mathbf{r}$, and $\Delta G^\circ(\mathbf{r}, \mathbf{s})$ is the free energy change of structure $\mathbf{s}$ for mRNA $\mathbf{r}$ according to an energy model. This is clearly a double minimization objective involving the per-sequence minimization over all of its possible structures (i.e., RNA folding; Eq. 2) which has well-known dynamic programming solutions, and the global minimization over all sequences (i.e., optimal mRNA design; Eq. 1) which we will solve using lattice parsing (§1.3).

Next, we integrate codon optimality by adding Codon Adaptation Index (CAI) (17), defined as the geometric mean of the codon optimality of each codon in the mRNA $\mathbf{r}$:

$$\text{CAI}(r) = \sqrt[\frac{|\mathbf{r}|}{3}]{\prod_{0 \le i < \frac{|\mathbf{r}|}{3}} w\big(\text{codon}(\mathbf{r}, i)\big)} \tag{3}$$

where $\text{codon}(\mathbf{r}, i) = r_{3i} r_{3i+1} r_{3i+2}$ is the ith triplet codon in $\mathbf{r}$, and $w(c)$ is the relative adaptiveness of codon c, defined as the frequency of c divided by the frequency of its most frequent synonymous codon ($0 \le w(c) \le 1$). Because CAI is always between 0 and 1 but MFE is generally proportional to the mRNA sequence length, we scale CAI by the number of codons and use a hyperparameter λ to balance MFE and CAI ($\lambda = 0$ being purely MFE), and define a novel joint objective:

$$\text{MFECAI}_\lambda(\mathbf{r}) = \text{MFE}(\mathbf{r}) - \frac{|\mathbf{r}|}{3} \lambda \log \text{CAI}(\mathbf{r}) \tag{4}$$

which can be simplified by expanding CAI:

$$\begin{aligned} \text{MFECAI}_\lambda(\mathbf{r}) &= \text{MFE}(\mathbf{r}) - \frac{|\mathbf{r}|}{3} \lambda \log \sqrt[\frac{|\mathbf{r}|}{3}]{\prod_{0 \le i < \frac{|\mathbf{r}|}{3}} w\big(\text{codon}(\mathbf{r}, i)\big)} \\ &= \text{MFE}(\mathbf{r}) - \lambda \sum_{0 \le i < \frac{|\mathbf{r}|}{3}} \log w\big(\text{codon}(\mathbf{r}, i)\big) \end{aligned} \tag{5}$$

This joint objective is basically MFE plus (a scaled) sum of the negative logarithm of each codon's

relative adaptiveness. Now the joint optimization can be defined as:

$$\mathbf{r}_\lambda^*(\mathbf{p}) = \text{argmin}_{\mathbf{r}\in mRNA(\mathbf{p})}\text{MFECAI}_\lambda(\mathbf{r})$$
$$= \text{argmin}_{\mathbf{r}\in mRNA(\mathbf{p})}\left(\text{MFE}(\mathbf{r}) - \lambda\sum_{0\le i<\frac{|\mathbf{r}|}{3}}\log w\big(\text{codon}(\mathbf{r},i)\big)\right) \quad (6)$$

See Fig. 2d for examples of relative adaptiveness calculation.

§1.2 DFA Representations for Codons and mRNA Candidate Sequences In- formally, a DFA is a directed graph with labeled edges and distinct start and end states. For our purpose each edge is labeled by a nucleotide, so that for each codon DFA, each start-to-end path represents a triplet codon; see Fig. **2**a and Extended Data Fig. 1c for examples. Formally, a DFA is a 5-tuple $\langle Q, \Sigma, \delta, q_0, F\rangle$, where Q is the set of states, Σ is the alphabet (here $\Sigma = \{\text{A,C,G,U}\}$), q_0 is the start state (always $(0,0)$ in this work), F is the set of end states (in this work the end state is unique, i.e., $F = \{(3,0)\}$), and δ is the transition function that takes a state q and a symbol $a \in \Sigma$ and returns the next state q', i.e., $\delta(q,a) = q'$ encodes a labeled edge $q \xrightarrow{a} q'$.

After building DFAs for each amino acid, we can concatenate them into a single DFA $D(\mathbf{p})$ for a protein sequence $\mathbf{p}$, which represents all possible mRNA sequences that translate into that protein

$$D(\mathbf{p}) = D(p_0)\circ D(p_1)\circ\cdots\circ D\big(p_{|\mathbf{p}|-1}\big)\circ D(\text{STOP})$$

by stitching the end state of each DFA with the start state of the next. See Extended Data Fig. 1d for examples. The new end state of the mRNA DFA is $(3|\mathbf{p}|+3,0)$.

We also define $out\_edges(q)$ to be the set of outgoing edges from state q, and $in\_edges(q)$ to be the set of incoming edges (which will be used in the pseudocode, see Supplementary Figs. 2–3):

$$out\_edges(q) = \{q \xrightarrow{a} q' \mid \delta(q,a) = q'\}$$
$$in\_edges(q) = \{q' \xrightarrow{a} q \mid \delta(q',a) = q\}$$

For the mRNA DFA in Extended Data Fig. 1d, $out\_edges\big((3,0)\big) = \{(3,0) \xrightarrow{\text{U}} (4,0), (3,0) \xrightarrow{\text{C}} (4,1)\}$ and $in\_edges\big((9,0)\big) = \{(8,0) \xrightarrow{\text{A}} (9,0), (8,0) \xrightarrow{\text{G}} (9,0), (8,1) \xrightarrow{\text{A}} (9,0)\}$.

§1.3 Objective 1 (Stability): Stochastic Context-Free Grammar, Lattice Parsing, and Intersection A stochastic context-free grammar (SCFG) is a context-free grammar in which each rule is augmented with a weight. More formally, an SCFG is a 4-tuple $\langle N, \Sigma, P, S\rangle$ where N is the set of non-terminals, Σ is the set of terminals (identical to the alphabet in the DFA, in this case $\Sigma = \{\text{A,C,G,U}\}$), P is the set of weight-associated context-free writing rules, and $S \in N$ is the start symbol. Each rule in P has the form $A \xrightarrow{w} (N \cup \Sigma)^*$ where $A \in N$ is a non-terminal that can be rewritten according to this rule into a sequence of non-terminals and terminals (the star $*$ means repeating zero or more times) and $w \in \mathbb{R}$ is the weight associated with this rule.

SCFGs can be used to represent the RNA folding energy model (39). The weight of a derivation (parse tree, or a secondary structure in this case) is the sum of weights of the productions used in that derivation. For example, for a very simple Nussinov-Jacobson-style model (40), which simplifies the energy model to individual base pairs, we can define this SCFG G as in Extended Data Fig. 1e, where each GC pair gets a score of -3, and each AU pair gets a score of -2. Thus, the standard RNA secondary structure prediction problem can be cast as a parsing problem: given the above SCFG G and an input RNA sequence, find the minimum-weight derivation in G that can generate the sequence. This can be solved by the classical CKY algorithm from computational linguistics (41,42,43).

The optimal-stability mRNA design problem is now a simple extension of the above single-sequence folding problem to the case of multiple inputs: instead of finding the minimum free energy structure (minimum weight derivation) for a given sequence, we find the minimum free energy structure (and its corresponding sequence) among all possible structures for all possible sequences (Extended Data Fig. 1). This can be solved by lattice parsing on the DFA, which is a generalization of CKY from a single sequence to a DFA. Take the bifurcation rule $S \rightarrow NP$ for example. In CKY, if you have derived non-terminal N for span $[i, j]$, notated $i \overset{N}{\underset{\Delta}{\rightarrow}} j$, and if you have also derived $j \overset{P}{\underset{\Delta}{\rightarrow}} k$, you can combine the two spans, i.e., $i \overset{N}{\underset{\Delta}{\rightarrow}} j \overset{P}{\underset{\Delta}{\rightarrow}} k$, and use the above rule to derive $i \overset{S}{\underset{\Delta}{\rightarrow}} k$. Similarly, in lattice parsing, if you have derived both $q_i \overset{N}{\underset{\Delta}{\rightsquigarrow}} q_j$ (i.e., there is a $q_i \rightsquigarrow q_j$ path that can be derived from N) and $q_j \overset{P}{\underset{\Delta}{\rightsquigarrow}} q_k$, you can combine them to a longer path $q_i \overset{N}{\underset{\Delta}{\rightsquigarrow}} q_j \overset{P}{\underset{\Delta}{\rightsquigarrow}} q_k$ and derive $q_i \overset{S}{\underset{\Delta}{\rightsquigarrow}} q_k$ with the above rule. While the runtime for CKY scales $O(|G|n^3)$ where $|G|$ is the grammar constant (the number of rules) and n is the RNA sequence length, the runtime for lattice parsing similarly scales $O(|G||D|^3)$ where $|D|$ is the number of states in the DFA. For mRNA design with the standard genetic code, $n \leq |D| \leq 2n$ because each position i has either one or two states ($(i, 0)$ and $(i, 1)$), so its time complexity is also actually identical to single-sequence folding, just with a larger constant. See Methods §1.6 for details of this algorithm and Supplementary Figs. 2–3 for the pseudocode.

More formally, in theoretical computer science, lattice parsing with an CFG G on a DFA D is also known as the intersection between the languages of G and D (i.e., the sets of sequences allowed by G and D), notated $L(G) \cap L(D)$, which was solved by the Bar-Hillel construction in 1961 (19). In order to adapt it to mRNA design, we need to extend this concept to the case of weighted (i.e., stochastic) grammars and weighted DFAs (the latter is needed for CAI integration; see below). While the language $L(G)$ of CFG G is the set of sequences generated by G, the language of the SCFG for RNA folding free energy model defines a mapping from each RNA sequence to its MFE, i.e.,$L_w(G)$: $\Sigma^* \longmapsto \mathbb{R}$. This can be written as a relation:

$$L_w(G) = \{\mathbf{r} \sim \text{MFE}(\mathbf{r}) \mid \mathbf{r} \in \Sigma^*\}$$

And we also extend the language of a DFA to a trivial weighted language (which will facilitate the

incorporation of CAI into DFA below):

$$L_w(D) = \{\mathbf{r}{\sim}0 \mid \mathbf{r} \in L(D)\}$$

Next we extend the intersection from two sets to two weighted sets A and B:

$$A \cap_w B = \{\mathbf{r}{\sim}(w_1 + w_2) \mid \mathbf{r}{\sim}w_1 \in A, \mathbf{r}{\sim}w_2 \in B\}$$

Now we can show that optimal-stability mRNA design problem can be solved via weighted intersection between $L_w(G)$ and $L_w(D)$, i.e., we can construct a new "intersected" stochastic grammar G' that has the same weights (i.e., energy model) as the original grammar but only generates sequences in the DFA:

$$L_w(G') = L_w(G) \cap_w L_w(D) = \{\mathbf{r}{\sim}\mathrm{MFE}(\mathbf{r}) \mid \mathbf{r} \in L(D)\}$$

§1.4 Adding Objective 2 (Codon Optimality): Weighted DFA for CAI Integration As described in the main text and Fig. 2d, our novel joint optimization objective (Eq. 6) factors the CAI of each mRNA candidate onto the relative adaptiveness of each of its codons, and thus can be easily incorporated into the DFA as edge weights. To do this we need to extend the definition of DFA to weighted DFA, where the transition function δ now returns a state and a weight, i.e., $\delta(q, a) = (q', w)$, which encodes a weighted label edge $q \xrightarrow{a:w} q'$. Now the set of outgoing and incoming edges are also updated to:

$$out\_edges(q) = \{q \xrightarrow{a:w} q' \mid \delta(q, a) = (q', w)\}$$
$$in\_edges(q) = \{q' \xrightarrow{a:w} q \mid \delta(q', a) = (q, w)\}$$

In this case, the weighted DFA defines a mapping from each candidate mRNA sequence to its negative logarithm of CAI scaled by the number of codons, i.e., $L_w(D)\colon L(D) \mapsto \mathbb{R}$. More formally,

$$L_w(D)\colon \{\mathbf{r}{\sim} -\tfrac{|\mathbf{r}|}{3}\log \mathrm{CAI}(\mathbf{r}) \mid \mathbf{r} \in L(D)\}$$

Now the weighted intersection defined above can be extended to incorporate the hyper-parameter λ and derive the joint objective:

$$L_w^{\lambda}(G') = L_w(G) \cap_w^{\lambda} L_w(D) = \{\mathbf{r}{\sim}(\mathrm{MFE}(\mathbf{r}) - \lambda\tfrac{|\mathbf{r}|}{3}\log \mathrm{CAI}(\mathbf{r})) \mid \mathbf{r} \in L(D)\}$$

§1.5 Bottom-Up Dynamic Programming Next, we describe how to implement the dynamic programming algorithm behind lattice parsing (or equivalently, intersection between the languages of a stochastic context-free grammar and a weighted DFA) to solve the joint optimization problem. For simplicity reasons, here we use bottom-up dynamic programming on a modified Nussinov-Jacobson energy model. Supplementary Fig. 2 gives the pseudocode for this simplified version. We first build up the mRNA DFA for the given protein, and initialize two hash tables, *best* to store the best score of each state, and *back* to store the best backpointer. For the base cases $\left(S \xrightarrow{0} N\ N\ N\right)$ we set $best[S, q_i, q_{i+3}] \leftarrow 0$ for optimal-stability design, and $best[S, q_i, q_{i+3}] \leftarrow \text{mincost}(q_i, q_{i+3}, \lambda)$ for the joint optimization where

$$\text{mincost}(q_i, q_{i+3}, \lambda) \triangleq \min_{q_i \xrightarrow{a:w_1} q' \xrightarrow{b:w_2} q'' \xrightarrow{c:w_3} q_{i+3}} \lambda\,(w_1 + w_2 + w_3) \tag{7}$$

is the minimum (λ-scaled) cost of any $q_i \rightsquigarrow q_{i+3}$ path in the CAI-integrated DFA. Next, for each state $\left(q_i, q_j\right)$ it goes through the pairing rule and bifurcation rules, and updates if a better score is found. After filling out the hash tables bottom-up, we can backtrace the best mRNA sequence stored with the backpointers. See Supplementary Fig. 3 for details of UPDATE and BACKTRACE functions.

§1.6 Left-to-Right Dynamic Programming Inspired by our previous work, LinearFold (21), we further developed a left-to-right dynamic programming, which is equivalent to the above bottom-up version but explores the search space incrementally from left to right; see Supplementary Fig. 4 for the pseudocode. This left-to-right order also enables beam search (44), a classical pruning technique, to significantly narrow down the search space without sacrificing too much search quality. Our real system uses this left-to-right dynamic programming on the Turner nearest neighbor free energy model (15,16), and our thermodynamic parameters follow LinearFold and Vienna RNAfold (45), except for the dangling ends, which do not contribute stability in LinearDesign. Dangling ends refer to stabilizing interactions for multiloops and external loops (46), which require knowledge of the nucleotide sequence outside of the state $\left(q_i, q_j\right)$. Though it could be integrated in LinearDesign, the implementation is more involved so we leave it to future work.

§1.7 DFAs for Other Genetic Codes, Coding Constraints, and Modified Nucleotides The DFA framework can also represent less common cases such as alternative genetic codes, modified nucleotides, and coding constraints. First, the DFA can encode non-standard genetic codes, such as alternative nuclear code for some yeast (47) and mitochondrial codes (48) (Extended Data Fig. 3a).

Second, we may want to avoid some unwanted or rare codons (such as the amber stop codon) which is an easy change on the codon DFAs (Extended Data Fig. 3b), or certain adjacent codon pairs that modulate translation efficiency (49), which is beyond the scope of single codon DFAs but easy on the mRNA DFA (Extended Data Fig. 3c). Similarly, we may want to disallow certain restriction enzyme recognition sites, which span across multiple codons (Supplementary Fig. 5). Finally, chemically modified nucleotides such as pseudouridine (Ψ) have been widely used in mRNA vaccines (38), which can also be incorporated in the DFA (Extended Data Fig. 3d).

§1.8 Related Work Here we first discuss the advantages of our algorithm over previous work, and then discuss a recent work (29) that uses LinearDesign in experimental screening.

Two previous studies (22,23) also tackled the problem of optimal-stability mRNA design (i.e., our objective 1) via dynamic programming, but their algorithms are complicated, not generalizable and less efficient. By contrast, the stability-only version of our work reduced the mRNA design problem to the classical computational linguistics problem of "lattice parsing", resulting in a much simpler and more efficient algorithm that is vastly different from the specifically-designed algorithms such as Cohen *et al.* (22) and CDSfold (23). More importantly, our work further solves the harder and practically more important problem of joint optimization between stability and codon optimality, which subsumes the stability-only objective as a special case. Here we comprehensively compare our work to the previous ones in the following seven aspects.

Lattice Representation of the Design Space Our work is the first to use automata theory to compactly and conveniently represent the exponentially large mRNA design space. By contrast, Cohen *et al.* and CDSfold extend the standard Zuker algorithm with the consideration of amino acid constraints, and they do not have any graph-theoretic or formal representation of the design space. To handle the nucleotide dependencies of the first and third positions in the codons of leucine and arginine, CDSfold introduces the "extended nucleotides", which classify the same nucleotide at the second position with different notations regarding the dependency. See Supplementary Fig. 6 for the lattice representation of leucine in our work as an example, and the extended nucleotides of leucine in CDSfold as a comparison. More importantly, our lattice representation is able to integrate (the logarithm) of CAI for a joint optimization of stability and codon optimality, and is general for arbitrary genetic code; see the details in later paragraphs.

Lattice Parsing Based on our DFA representation, we further reduce the mRNA design problem to the classical computational linguistics problem of lattice parsing, which aims to find the most grammatical sentence among exponentially many alternatives. This problem was solved by Bar-Hillel *et al.* in 1961 (19). Therefore, instead of inventing a new algorithm, we simply adapt the classical lattice parsing to mRNA design using our algorithm of LinearDesign. Note that the single-sequence folding is a special case of our algorithm where the lattice is a single chain.

Efficiency More interestingly, our simple adapted algorithm reduces the constant factor of the

cubic-time bifurcation rule that dominates the runtime of mRNA design, leading to better efficiency over previous work such as CDSfold. Supplementary Fig. 7b illustrates the space and time complexity under the classical Nussinov energy model.

The single-sequence RNA folding defines a span (i, j) as an item, where i and j are indices in the RNA sequence. For a sequence with n nucleotides, during dynamic programming, at most n^3 items are generated for the bifurcation rule $S \longrightarrow S\ P$; space-wise, at most n^2 items are stored.

Extending RNA folding to lattice parsing, our work defines each item as (q_i, q_j), where q_i and q_j are the nodes in the lattice: $q_i \in \{(i, 0), (i, 1)\}$ and $q_j \in \{(j, 0), (j, 1)\}$. Since there are at most two nodes at each position, the number of items stored is at most $4n^2$. For the bifurcation rule $S \to SP$, items (q_i, q_k) and (q_k, q_j) are combined to form a bigger item (q_i, q_j), in which at most $8n^3$ items are generated (at most 2 nodes each for i, k and j). See Supplementary Fig. 7c for the illustration of above analysis; see lines 22–25 in Supplementary Fig. 2 and lines 20–24 in Supplementary Fig. 4 for the pseudocode of the bifurcation case in our work.

By contrast, CDSfold defines each item as (i, j, nuc_i, nuc_j), where nuc_i and nuc_j are the nucleotides at positions i and j, respectively. The number of items stored in CDSfold scales $16n^2$, because there are at most 4 nucleotide types for each nuc_i and nuc_j. For the bifurcation rule $S \to S\ P$, items (i, k, nuc_i, nuc_k) , and $(k+1, j, nuc_{k+1}, nuc_j)$, are combined to form (i, j, nuc_i, nuc_j), in which at most $128n^3$ items are generated (at most 4×4 nucleotide types at nuc_i and nuc_j, and 4×2 nucleotide pairs between nuc_k and nuc_{k+1}). See Supplementary Fig. 7d for the analysis illustration of CDSfold.

Compared to CDSfold, our work largely reduces the time complexity constant of the bifurcation rule $S \longrightarrow S\ P$ from 128 to 8. The cubic-time bifurcation rule which dominates the runtime in CDSfold is greatly accelerated in our algorithm. Empirically, our algorithm scales quadratically rather than cubically with mRNA sequence length for practical applications (Fig. 3 & Supplementary Fig. 8).

Joint Optimization of Stability and Codon Optimality Codon optimality is an important factor in mRNA design, which should be jointly optimized with stability (5), and our work is the first to solve this joint optimization problem, thanks to the DFA representation and lattice parsing. By contrast, previous work (Cohen *et al*. and CDSfold) does not perform, and is impossible to be extended to perform, such a joint optimization:

- Firstly, Cohen *et al.* only optimize stability without considering codon optimality. CDSfold uses simulated annealing to improve CAI by fine-tuning from the MFE solution, but this is a heuristic with no guarantees.

- Secondly, CDSfold's objective function, $\mathrm{MFE} \cdot \mathrm{CAI}^{\lambda}$, is impossible for dynamic programming due to the difference between MFE and CAI, where MFE is a sum of free energy for each component substructure (additive) but CAI is a geometric mean of the

relative codon usages (multiplicative). To reconcile this difference, our formulation defines a novel objective that factors the logarithm of CAI for an mRNA *additively* onto its individual codons, thus making it decomposable and amenable to dynamic programming (see Methods §1 for details). By contrast, CDSfold's objective formulation does not factor into individual codons, and thus cannot be incorporated into global optimization.

- Last but most importantly, even if CDSfold were to borrow our formulation, its fundamental codon representation still rules out joint optimization. Our framework easily encodes (the logarithm of) CAI in our DFA representation, for example, we can integrate CAI onto a weighted DFA for leucine (Supplementary Fig. 6a). By contrast, CDSfold has to use an "extended nucleotides" representation for codon choices, which makes it impossible to do joint optimization with CAI (Supplementary Fig. 6b–c).

To summarize, our framework easily incorporates codon optimality into the joint optimization that previous work did not (and could not be extended to) tackle. Our objective integrates (the logarithm of) CAI and MFE together, while the objective of CDSfold is not able to reconcile these two factors. Furthermore, even if using our objective formulation, CDSfold's representation of codon choices still rules out the possibility of CAI integration.

Generalizability Our DFA framework is so general that it can also represent arbitrary (non-standard) genetic codes, modified nucleotides, and coding constraints such as adjacent codon pair preference, which previous work could not handle even with major modifications. See Methods §1.7 for details.

Linear-time Version for Long Sequence and Suboptimal Candidates We further develop a faster, linear-time, approximate version which greatly reduces runtime for long sequences with small sacrifices in search quality, which we also use to generate multiple suboptimal candidates with varying folding stability and codon optimality as candidates for experimentation.

Verification of Wet Lab Experiments Extensive experiments confirm that compared to the standard codon optimization benchmark, our designs are substantially better in chemical stability and protein expression *in vitro*, and the corresponding mRNA vaccines elicit up to 128×higher antibody responses *in vivo*.

Another recent work (29) optimized mRNA designs and screened them via an experimental platform. LinearDesign played a central role in their work as the starting point of their optimizations (see Fig. 4b of their paper), followed by fine-tunings by both human players and a Monte Carlo tree search algorithm. The resulting coding regions are flanked by different UTRs, and then tested on stability and protein expression. LinearDesign-generated sequences showed strong stability and protein expression results with different UTRs (Figs. 2g and 4a of their paper), independently confirming our *in vitro* experiments. However, they did not perform any *in vivo* validations.

§1.9 Benchmark Dataset and Machine To estimate the time complexity of LinearDesign, we collected 114 human protein sequences from UniProt (24), with length from 78 to 3,333 amino acids (not including the stop codon); see Supplementary Tab. 1. We benchmarked LinearDesign on a Linux machine with 2 Intel Xeon E5-2660 v3 CPUs (2.60 GHz) and 377 GB memory, and used Clang (11.0.0) to compile. The code only uses a single thread.

§1.10 Additional Design Constraints Some studies have shown that protein expression level drops if the 5'-end leader region has more secondary structure (50,51,52,53,5). To design sequences with less structures at 5'-end leader region, we take a simple "design, enumerate and concatenate" strategy to avoid structure in the leader region: (1) design the CDS region except for the 5'-end leader region (i.e., the first 15 nucleotides); (2) enumerate all possible subsequences in the 5'-end leader region; and (3) concatenate each subsequence with the designed sequence, refold, and choose the one whose 5'-end leader region has the most unpaired nucleotides.

In addition, it has been revealed that long double-stranded regions may induce unwanted innate immune responses by previous studies (27,54,55). Considering this, we do not allow long double-stranded regions that include 33 or more base pairs by adding this constraint in the design process.

§1.11 RNA Secondary structure prediction and visualization Vienna RNAfold from ViennaRNA package (version 2.4.14) is used for predicting and drawing the secondary structure of mRNA sequence, and calculating the Minimum Free Energy (MFE) of secondary structures.

§2 Details of *In vitro* and *In vivo* Experiments

§2.1 Preparation of mRNA and its formulation mRNA molecules were synthesized *in vitro* by T7 RNA polymerase using linearized plasmid as DNA template. The open-reading frame (ORF) region is flanked with the 5' and 3' untranslated regions (UTRs) followed by a 70 *nt* poly-A tail. For all Spike protein-coding sequences, the *in vitro* transcription reaction was conducted at 37 °C for 4 hours, followed by digestion with DNase I (Hongene Biotech). mRNA encoding full-length S protein without proline substitution was then capped using Vaccinia Capping Enzyme (Hongene Biotech) and purified with magnetic Dynabeads (Thermo Fisher). Eluted mRNA was further treated with Antarctic Phosphatase (Hongene Biotech) at 37 °C for 30 minutes to remove residual 5'-triphosphates. For all VZV gE sequences, mRNA was co-transcriptionally capped using m7(3'OMeG)(5')ppp(5')(2'OMeA)pG capping reagent (Hongene Biotech) in a "one-pot" reaction at 30 °C for 16 hours, followed by treatment with DNase I. Capped mRNA encoding Spike or VZV gE protein was then purified using beads. For the preparation of formulated mRNA vaccines, lipopolyplex (LPP) formulation was used to encapsulate mRNA cargo as described previously (56). LPP is a lipid-based mRNA delivery system and has been demonstrated to provide high efficacy and good safety profile (26).

§2.2 Agarose gel electrophoresis and integrity assay of mRNA To study the electrophoretic mobility profile of mRNA molecules, mRNA samples suspended in Am- bion® RNA storage buffer (Thermo Fisher) were denatured at 75 °C for 5 minutes and snap – cooled on ice before loaded onto 1% non-denaturing agarose gel (130 V for 1 h at room temperature). Gel image was taken by Gel Doc XR+ Gel Documentation System (Bio-Rad).

To assess the in-solution stability of mRNA, samples were incubated in PBS buffer containing 10 mM Mg^{2+}. Sampling was conducted at time points (0, 1, 2, 4, 8, 12, 16, 24, 32, 48, and 60 hours). For a faster degradation process, PBS buffer containing 20 mM Mg^{2+} instead of 10 mM was used. Sampling was done in a relatively shorter time span (0, 1, 2, 4, 8, 12, 15, 18, 21, and 24 hours). RNA integrity was analyzed by Qsep100™ Capillary Electrophoresis System. The integrity was represented as the proportion of full-length mRNA calculated on electropherogram. The data were normalized to time point 0 h. To extrapolate the half-life of each sequence, one-phase decay equation:

$$Y = (Y_0 - \text{Plateau}) \cdot e^{-KX} + \text{Plateau}$$

was used to fit the curve. The Y_0 and Plateau were set as 100 and 0, respectively. Half-life was computed as $ln(2)/K$, where K refers to decay rate constant.

§2.3 Protein expression assay Human embryonic kidney 293 cells (HEK293) (ATCC) were cultured in Dulbecco's Modified Eagle's Medium (DMEM) (Hyclone) containing 10% fetal bovine serum (FBS) (GEMINI) and 1% Penicillin-Streptomycin (Gibco). All cells were cultured at 37 °C in a 5% CO_2 condition.

For the measurement of protein expression, cells were transfected with mRNA using Lipofectamine MessengerMAX (Thermo Scientific). Briefly, a mix of 2 μg mRNA and 6 μL of Lipofectamine reagent was prepared following the manual instructions and then incubated with cells for 24 or 48 hours. For flow cytometric analysis, cells were collected and stained with live/dead cell dye (Fixable Viability Stain 510, BD) for 5 min. After washing, cells were incubated with anti-RBD chimeric mAb (1:100 dilution, Sino Biological) for 30 min, followed by washing and incubation with PE-anti-human IgG Fc (1:100 dilution, Biolgend) for 30 min. Samples were analyzed on BD Canto II (BD Biosciences). Data were processed using FlowJo V10.1 (Tree Star).

§2.4 *In vivo* immunogenicity study C57BL/6 mice (6-8 weeks) were intramuscularly immunized twice with 10 μg LPP formulated mRNA vaccines at a 2-week interval. Sera and spleens were collected 14 days after boost shot.

Surrogate Virus Neutralization (sVNT) Assay Neutralizing antibody titer was measured using sVNT assay as previously described (57) with some modifications. Briefly, 96-well plates (Greiner Bio-one) were coated recombinant with hACE2 protein (100 ng/well, Genscript) overnight at 4 °C. Plates were washed with 1×PBS-T and blocked with 2% BSA for 2 hours at RT. HRP-conjugated RBD (100 ng/mL) were incubated with serially diluted serum from immunized mice at an equal volume (60 μL each) for 30 min at 37 °C. Sera collected from PBS-treated mice were used

as negative control. Then a 100 μL mixture of RBD and serum was added into each well and incubated for 15 min at 37 °C. After washing, TMB substrate (Invitrogen) was used for color development and the absorbance at 450 nm was recorded using BioTek microplate reader. The IC50 value was calculated using 4 parameter logistic non-linear regression.

Enzyme-Linked ImmunoSorbent Assays (ELISA) Briefly, recombinant SARS- CoV-2 Spike ectodomain protein or VZV gE protein (Genscript) diluted in coating buffer (Biolegend) were used to coat 96-well EIA/RIA plates (Greiner Bio-one, 100 ng/well) at 4 °C overnight. The plates were then washed with 1×PBS-T (0.05% Tween-20) and blocked with 2% BSA in PBS-T for 2 hours at RT. Serum samples with serial dilutions were added and incubated for 2 hours at RT. After washing, HRP-conjugated goat anti-mouse IgG Ab (1:10,000) was added and incubated for 1 hour. TMB substrate (Invitrogen) was then used for color development and the absorbance was read at 450 nm using BioTek microplate reader. Endpoint titers were calculated as the largest sample dilution factor yielding a signal that exceeds 2.1-fold value of the background (58).

Enzyme-linked Immunospot (ELISpot) Assay Frequency of Spike (or VZV gE) antigen-specific IFN-γ-secreting T cells was evaluated using Mouse IFN-γ ELISpotplus Kit (Mabtech) according to the manual. Briefly, 3×105 murine splenocytes were added to wells pre-coated with anti-mouse IFN-γ capturing Abs and were incubated with Spike protein or VZV gE peptide pool (10 μg/mL) for 20 hours. After washing, plates were incubated with Streptavidin-ALP (1:1000) for 1 hour at RT. Spots were developed with BCIP/NBT substrate solution and counted using Immunospot S6 analyzer (CTL). Due to multiple steps and exponential change of antibody and antigen-specific T cells during the immunity induction process, *in vivo* immunogenicity data usually have high data variations. Inoculation of mRNA vaccine involves extra processes such as tissue transfection and protein translation, and the variations in these process efficiencies together with variable dosing and differences in individual mouse's immune status usually bring more immunogenicity variations than protein-based vaccines. From our experience, the variations observed in this study are typical for mRNA vaccines.

§2.5 Ethics statement All mouse studies were performed in strict accordance with the guidelines set by the Chinese Regulations of Laboratory Animals and Laboratory Animal-Requirements of Environment and Housing Facilities. Animal experiments were carried out with the approval from the Institutional Animal Care and Use Committee (IACUC) of Shanghai Model Organisms Center, Inc.

§2.6 Statistics and Reproducibility Geometric means or arithmetic means are represented by the heights of bars, or symbols, and error bars represent the corresponding s.d. Two-tailed Mann–Whitney U tests were used to compare two experimental groups for the *in vivo* studies. To compare more than two experimental groups, One-way ANOVA with Dunn's multiple comparisons tests were applied in the *in vitro* protein expression experiment. Statistical analyses were performed

using Prism v.8 (GraphPad). *$p < 0.05$, **$p < 0.01$, ***$p < 0.001$. The raw p values from the statistical analysis were summarized in the figshare file. *In vitro* experiments were independently repeated in triplicate. Animal experiments were completed once. Gel electrophoresis experiment was repeated three times to obtain similar results.

§2.7 Data reporting No statistical methods were used to predetermine sample size. The experiments were not randomized. The investigators were not blinded to allocation during experiments and outcome assessment.

39. Rivas E (2013) The four ingredients of single-sequence RNA secondary structure prediction. a unifying perspective. RNA Biology 10(7):1185–1196.
40. Nussinov R, Jacobson AB (1980) Fast algorithm for predicting the secondary structure of single-stranded RNA. Proceedings of the National Academy of Sciences U.S.A. 77(11):6309–6313.
41. Kasami T (1966) An efficient recognition and syntax-analysis algorithm for context-free languages. Coordinated Science Laboratory Report no. R-257.
42. Younger DH (1967) Recognition and parsing of context-free languages in time n3. Information and Control 10(2):189–208.
43. Rivas E, Lang R, Eddy. R (2012) A range of complex probabilistic models for RNA secondary structure prediction that includes the nearest-neighbor model and more. RNA 18(2):193–212.
44. Huang L, Fayong S, Guo Y (2012) Structured perceptron with inexact search in Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. (Association for Computational Linguistics, Montréal, Canada), pp. 142–151.
45. Lorenz R, et al. (2011) ViennaRNA package 2.0. Algorithms for Molecular Biology 6(1):1.
46. Turner DH, Mathews DH (2010) NNDB: the nearest neighbor parameter database for predicting stability of nucleic acid secondary structure. Nucleic Acids Research 38(suppl_1):D280–D282.
47. Kawaguchi Y, Honda H, Taniguchi-Morimura J, Iwasaki S (1989) The codon CUG is read as serine in an asporogenic yeast candida cylindracea. Nature 341(6238):164–166.
48. Bonitz SG, et al. (1980) Codon recognition rules in yeast mitochondria. Proceedings of the National Academy of Sciences U.S.A. 77(6):3167–3170.
49. Gamble CE, Brule CE, Dean KM, Fields S, Grayhack EJ (2016) Adjacent codons act in concert to modulate translation efficiency in yeast. Cell 166(3):679–690.
50. Ding Y, et al. (2014) *In vivo* genome-wide profiling of RNA secondary structure reveals novel regulatory features. Nature 505(7485):696–700.
51. Wan Y, et al. (2014) Landscape and variation of RNA secondary structure across the human transcriptome. Nature 505:706–709.
52. Shah P, Ding Y, Niemczyk M, Kudla G, Plotkin JB (2013) Rate-limiting steps in yeast protein translation. Cell 153:1589–601.
53. Tuller T, Zur H (2014) Multiple roles of the coding sequence 5' end in gene expression regulation. Nucleic Acids Research 43(1):13–28.
54. Husain B, Mukerji I, Cole JL (2012) Analysis of high-affinity binding of protein kinase R to double-stranded RNA. Biochemistry 51(44):8764–8770.
55. Hur S (2019) Double-stranded RNA sensors and modulators in innate immunity. Annual Review of Immunology 37:349–375.
56. Persano S, et al. (2017) Lipopolyplex potentiates anti-tumor immunity of mRNA-based vaccination. Biomaterials 125:81–89.
57. Tan CW, et al. (2020) A SARS-CoV-2 surrogate virus neutralization test based on antibody-mediated blockage of ACE2–spike protein–protein interaction. Nature Biotechnology 38(9):1073–1078.

58. McKay PF, et al. (2020) Self-amplifying RNA SARS-CoV-2 lipid nanoparticle vaccine candidate induces high neutralizing antibody titers in mice. Nature communications 11(1):3523.

Data Availability

The UniProt sequences used to estimate the time complexity of LinearDesign are included in Supplementary Tab. 1 and deposited at our figshare repository https://doi.org/10.6084/m9.figshare.22193251. The COVID-19 and VZV mRNA coding region sequences and UTR sequences used in the biological experiments are included at the end of Supplementary Information file and available on our figshare repository. Source data of the animal experiments is provided with this paper, and all source data of wet lab experiments is available on that repository.

Code Availability

The LinearDesign source code is available to all parties on GitHub (https://github.com/LinearDesignSoftware/LinearDesign) and Zenodo (https://doi.org/10.5281/zenodo.7839739), and is free for academic and research use.

Competing Interests

Baidu USA filed a patent for the LinearDesign algorithm in 2021 listing H.Z., L.Z., Z.L., K.L., B.L., and L.H. as inventors. The work of H.Z., L.Z., Z.L., K.L., B.L., and L.H. for the development of the LinearDesign algorithm was conducted at Baidu USA. StemiRNA Therapeutics has filed a provisional patent for the VZV mRNA vaccine listing C.X., H.S., and H.L. as inventors. Sanofi entered a non-exclusive licensing agreement with Baidu USA in 2021 to use LinearDesign to develop mRNA vaccines and therapeutics. H.Z., L.Z., Z.L., K.L., B.L., and L.H. is/were employees of Baidu USA. A.L., C.X., X.M., F.Z., H.J., C.C., H.S., H.L. and Y.Z. are/were employees of StemiRNA Therapeutics. L.H. and D.H.M. are also cofounders of Coderna.ai, Inc.

Author Contributions

L.H. conceived and directed the project. L.H. designed the basic algorithm for the Nussinov model and wrote a Python prototype, and H.Z. and L.Z. extended this algorithm to the Turner model, and implemented it in C++, which Z.L. optimized. L.H., H.Z., and L.Z. designed the CAI integration algorithm which L.Z. and H.Z. implemented. L.Z. implemented the beam search and handled design constraints. K.L. made the web server. B.L. implemented a baseline. Y.Z. and H.L. supervised the *in vitro* and *in vivo* experiments. C.C. performed the mRNA synthesis and gel electrophoresis experiments. A.L., C.X., H.J., X.M., F.Z. performed the protein expression and *in vivo* assays, and C.X. performed chemical stability and structure compactness assays. D.H.M. discussed the approach and provided guidance for *in silico* analysis and writing. L.H., H.Z., L.Z., D.H.M., A.L., C.X., H. S., H.L. and Y.Z. wrote the manuscript.

Acknowledgments

We thank Rhiju Das (Stanford) for introducing the mRNA design problem to us, Robin Li (Baidu) for connecting Baidu Research with StemiRNA, Julia Li (Baidu Research) for coordinating resources for this project, Goro Terai and Kiyoshi Asai (Univ. of Tokyo) for sending us the

CDSfold code, Sharon Aviran (UC Davis) for spotting a typo in the hyper-parameter λ in our earlier version, Alicia Solórzano (Pfizer) for the question on LinearDesign's independence of the choice of UTRs, Jinzhong Lin (Fudan) for early discussions, and Sizhen Li (Oregon State Univ.) for proofreading and help on LATEX. We acknowledge the assistance from Lei Huang and Mingyang Liu (StemiRNA) in LPP formulation of mRNA vaccines and help from other StemiRNA's colleagues including Yinglei Yi, Qiuhe Wang, Weiyun Wang and Yun Ge with *in vivo* studies. We thank Sanofi and many other vaccine companies worldwide for licensing and early adoption of LinearDesign. D.H.M. is supported by National Institutes of Health grant R35GM145283. A.L. was StemiRNA's employee and is currently supported by the Natural Science Foundation of Jiangsu Province (BK20221031), the National Science Foundation of China (32200764, 82061138008) and the Fundamental Research Funds for the Central Universities (2632022YC01). C.X. was sponsored by Shanghai Pujiang Talent Program (22PJ1423100). The funding from StemiRNA Therapeutics were supported by the Science and Technology Commission of Shanghai Municipality, China (20S11909100, 22S11902300); Shanghai Strategic Emerging Industry Development special fund (ZJ640070216) and the project of mRNA Innovation and Translation Center, Shanghai, China.

Extended Figures and Tables

Extended Data Figure 1: Illustrations of the optimization problems in mRNA design, DFA representations, single sequence folding as natural language parsing, and lattice parsing. a–b, Visualization of mRNA design as optimization problems for stability (objective 1, in **a**) and joint stability and codon optimality (objectives 1 & 2, in **b**). **c–h** show how lattice parsing solves the first optimization problem (see Fig. 2D for the second). **c,** Codon DFAs. **d,** An mRNA DFA made of three codon DFAs. The thick paths depict the optimal mRNA sequences under the simplified energy model in **e**, AUGCU⋆UGA, where ⋆ could be any nucleotide. **e,** Stochastic context-free grammar (SCFG) for a simplified folding free energy model. Each rule has a cost (i.e., energy term, the lower the better), and the dotted arcs represent base pairs in RNA secondary structure. **f,** Single-sequence folding is equivalent to context-free parsing with an SCFG; the parse tree represents the best secondary structure for the input mRNA sequence. **g,** We extend single-sequence parsing (top) to lattice parsing (bottom) by replacing the input string with a DFA, where each string index becomes a DFA state, and a span becomes a path between two states. **h,** Lattice parsing with the grammar in **e** for the DFA in **d**. The blue arcs below the DFA depict the (shared) best structure for the optimal sequences AUGCU⋆UGA in the whole DFA, while the dashed light-blue arcs above the DFA represent the best structure for a suboptimal sequence AUGUUAUAA. Lattice parsing can also incorporate codon optimality (objective 2, see **b**), by replacing the DFA with a weighted one (Fig. 2d).

Extended Data Figure 2: Word lattice and lattice parsing in natural language processing, and correspondence between linguistics and biology. **a,** An example of word lattice (sentence DFA) for speech recognition. **b,** Simplified language grammar. **c,** Single sentence parsing with between-word indices, which is a special case of word lattice parsing. **d,** Illustration of word lattice parsing for speech recognition with given word lattice and language grammar; the dashed blue arcs above the DFA depict the best parsing structure for the optimal sentence "I like this meal", while the dashed light-blue arcs below the DFA represent the best parsing structure for a non-optimal sentence "alike this veal". **e,** Correspondence between computational linguistics (left) and computational biology (right). See also Fig. 1.

Extended Data Figure 3: Examples of the DFA representations for extended codons, modified nucleotides, and coding constraints. a, Alternative genetic codes of serine, tryptophan, and threonine. **b,** Avoiding certain codon. On the left it shows the original DFA of serine (up), in which the red dashed arrows indicating UCA and UCG are chosen to be avoided, resulting in a new DFA (down). On the right it shows removing the rare amber STOP codon (UAG). **c,** Avoiding a specific adjacent codon pair. **d,** Extended serine DFA can include chemically modified nucleotides pseudouridine (Ψ), 6-Methyladenosine (m^6A) and 5-methylcytosine (m^5C).

Extended Data Figure 4: Two dimensional (MFE-CAI) visualizations of mRNA designs for the Spike protein using human codon preference (a) and yeast codon preference (b) with positive and negative λ's. GC% are shown in parentheses. The human genome prefers GC-rich codons that lead to higher CAI designs are with higher GC%, while the yeast genome prefers AU-rich codons that exhibit an opposite relationship between CAI and GC%. See also Fig. 3 for more *in silico* results of LinearDesign.

Extended Data Figure 5: Extra experimental results of LinearDesign-generated mRNAs encoding the Spike protein. a, In-solution stability of sequences A–H in PBS buffer containing 20 mM Mg^{2+} at 37 °C over the course of 24 h. The degradation experiments were performed in triplicate independently and the data were presented as mean ± s.d. and fitted with a one-phase decay curve. **b**, Protein expression of mRNAs following transfection into HEK293 cells for 24 hours was determined by flow cytometry. MFI values derived from three independent experiments are shown. Kruskal–Wallis analysis of variance (ANOVA) with Dunn's multiple comparisons test to H group was performed for statistical analysis.. ***$p < 0.001$. See Fig. 4c-d for similar experiments but with 10 mM Mg^{2+} and 48 hours, respectively.

Extended Data Figure 6: In-solution stability and protein expression of sequences A, C, H and BNT. a-b, In-solution stability of mRNAs in PBS buffer containing 20 mM Mg^{2+} or 10 mM Mg^{2+} at 37 °C. Data are from three independent experiments and were presented as mean ± s.d. and fitted with one-phase decay curve. **c-d**, Protein expression of mRNAs was determined 24 hours or 48 hours following transfection into HEK293 cells. MFI value is presented as mean ± s.d. Each group has three independent assays and 10,000 live cells were collected for analysis in each assay. Kruskal–Wallis analysis of variance (ANOVA) with Dunn's multiple comparisons test to BNT group was performed for statistical analysis. **$p < 0.01$, ***$p < 0.001$.

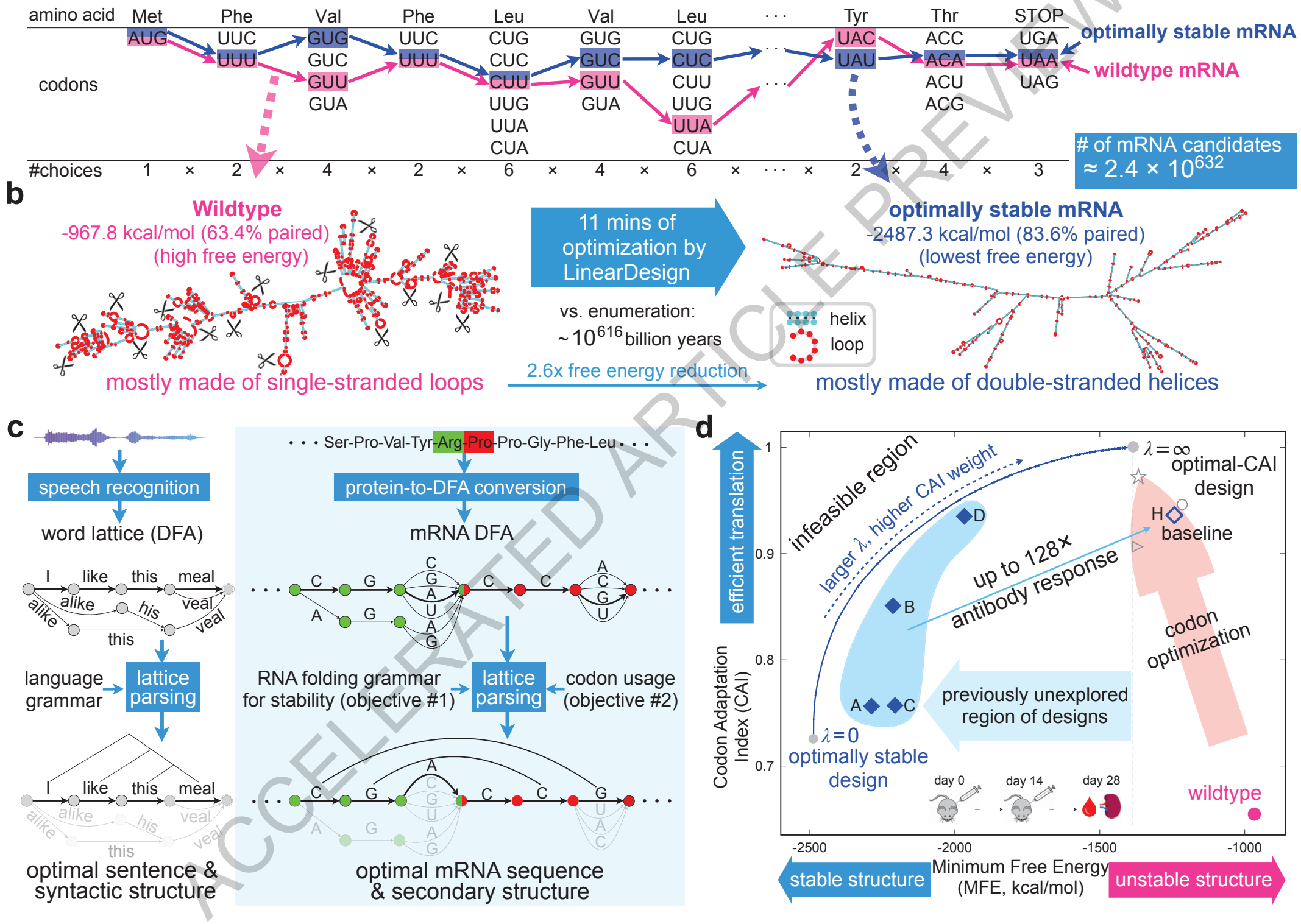
Extended Data Figure 7: Antibody (Ab) responses induced by sequences A, C, H and BNT-based mRNA vaccines. C57BL/6 mice (n=5) were immunized *i.m.* with two doses of mRNA vaccines at a 2-week interval. Seven days after boost immunization, levels of anti-Spike IgG (**a**) and neutralizing Abs (**b**) against pseudotyped SARS-CoV-2 were measured. Data were presented as geometric mean ± geometric s.d. A two-tailed Mann-Whitney U test was used for statistical analysis. *$p < 0.05$. See Source Data for details.

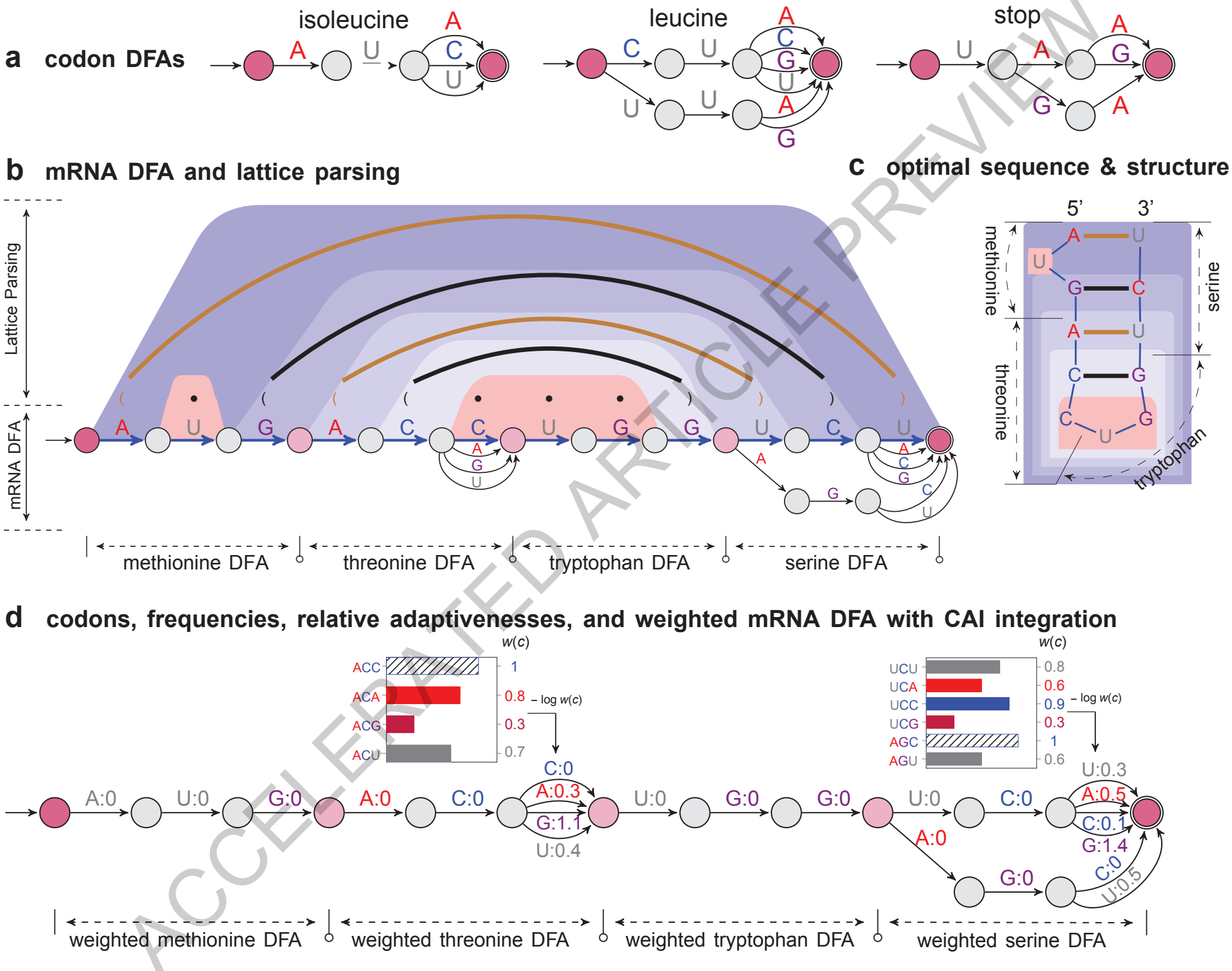
Extended Data Figure 8: Extra stability and protein expression results of LinearDesign-generated mRNAs encoding VZV gE protein. a, In-solution stability of mRNAs upon incubation in buffer (Mg^{2+} = 20 mM) at 37 °C. Percentage of intact mRNA is shown. Data are presented as mean ± SD from three independent experiments. **b**, Protein expression of mRNAs following transfection into HEK293 cells for 24 hours was determined by flow cytometry. Each group has three independent assays and 10,000 live cells were collected for analysis in each assay. MFI value is presented as mean ± s.d. Kruskal–Wallis analysis of variance (ANOVA) with Dunn's multiple comparisons test to gE-Ther group was performed for statistical analysis. *$p < 0.05$, ***$p < 0.001$.

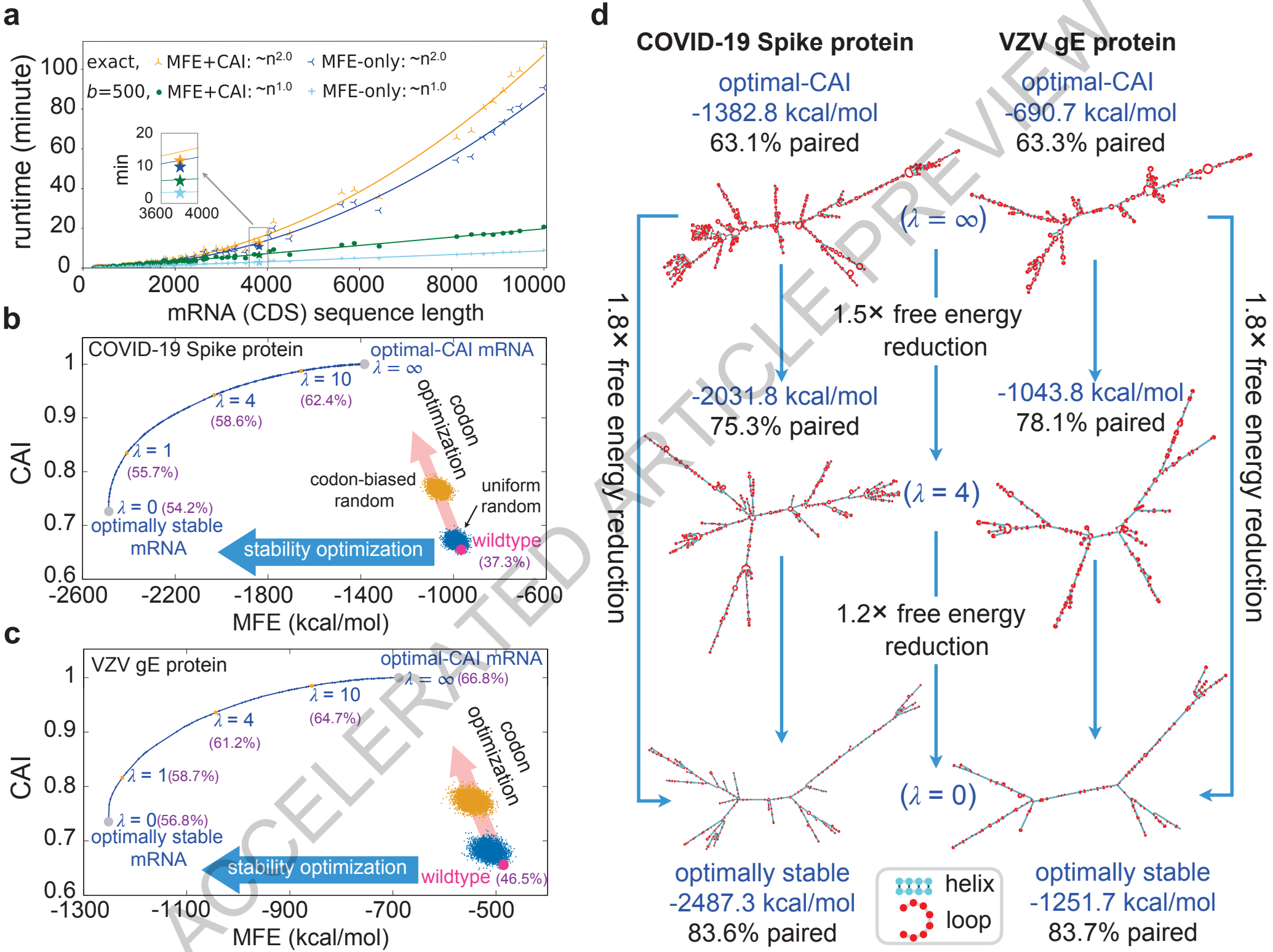
Extended Data Table 1: The LinearDesign-generated coding-region sequences, due to more secondary structures within the coding region, are less likely to form base pairs with or interfere with the structures of the UTRs. Here we show the numbers of predicted base pairs between UTRs and the coding regions of SARS-CoV-2 Spike protein. We used 5 different UTRs: StemiRNA COVID-19 UTRs used in wet lab experiments, BNT162b2 (BioNTech) UTRs, mRNA-1273 (Moderna) UTRs, CV2CoV

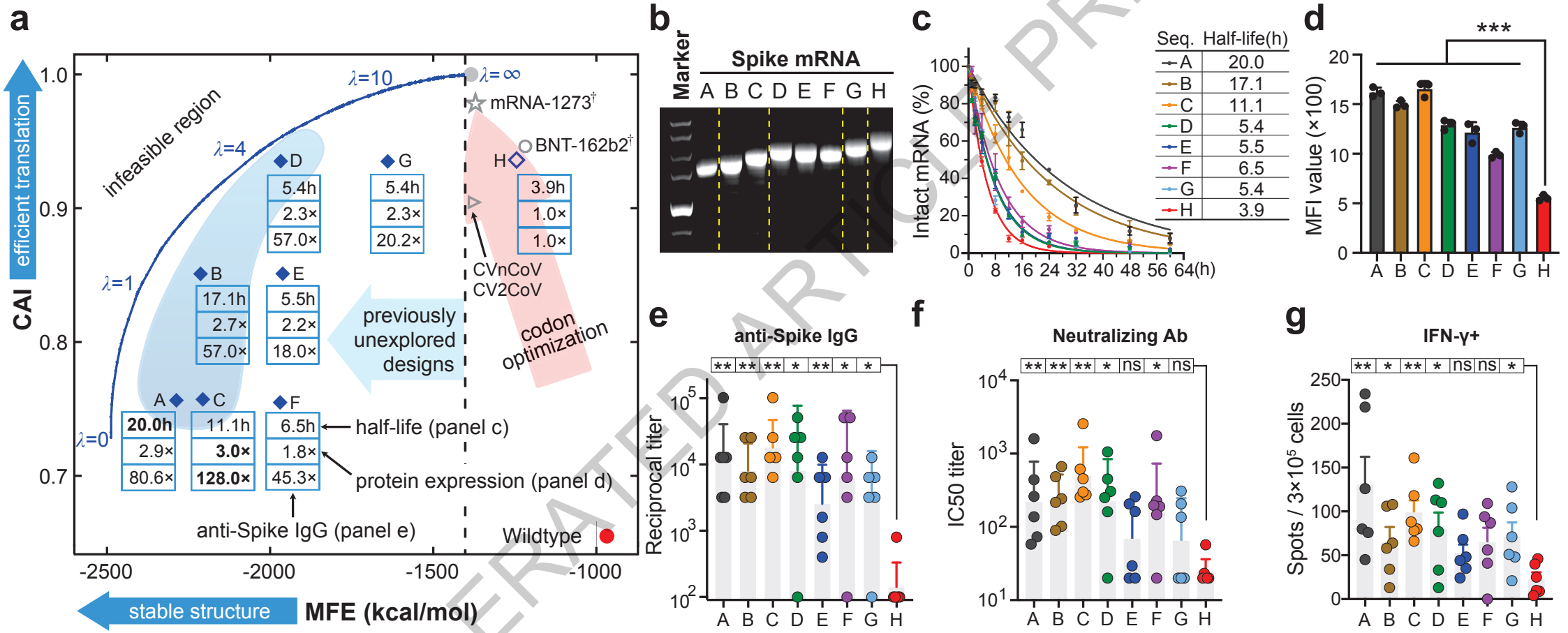
(CureVac) UTRs, and a widely used human β-globin mRNA UTRs. We tested 14 different sequences of the coding region: sequences A–H for wet lab experiments, sequences from three main mRNA vaccine companies, MFE-opt. and CAI-opt. sequences (i.e., sequences with the lowest folding free energy and with CAI=1, respectively), and the wildtype sequence. Most of the LinearDesign-generated mRNA sequences (sequences A–F and MFE-opt.) form fewer base pairs with UTRs. The folding free energies and structures are predicted by Vienna RNAfold (-d0 mode); MFEs of CDS are calculated without stop codon.

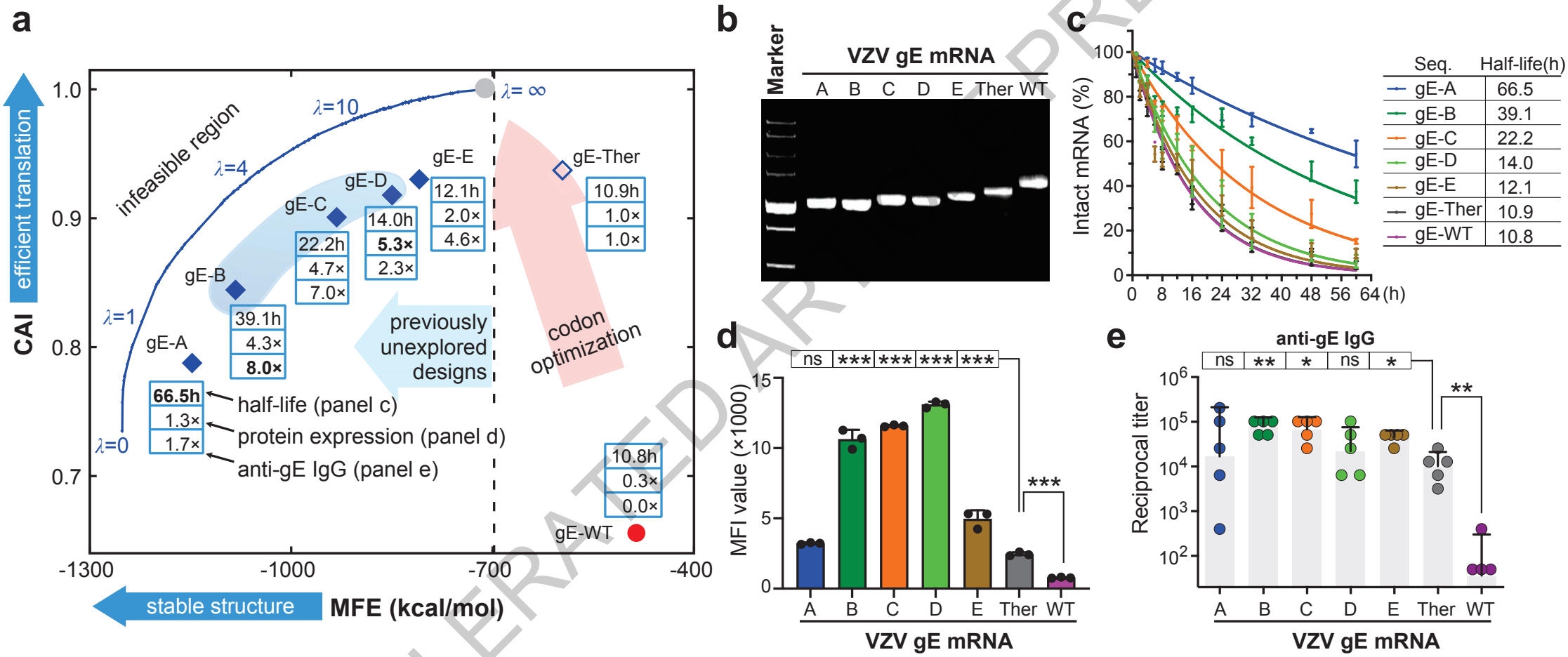
Extended Data Table 2: Similar to Extended Data Table 1, LinearDesign-generated coding-regions for the VZV gE protein form less base pairs with the UTRs. Here we used 5 different UTRs: StemiRNA VZV UTRs used in wet lab experiments, BNT162b2 (BioNTech) UTRs, mRNA-1273 (Moderna) UTRs, CV2CoV (CureVac) UTRs, and human β-globin mRNA UTRs. The 7 coding sequences are gE A–E, gE-Ther and gE-WT, which are used in the wet lab experiments of VZV. The folding free energies and structures are predicted by Vienna RNAfold (-d0 mode); MFEs of CDS are calculated without stop codon.

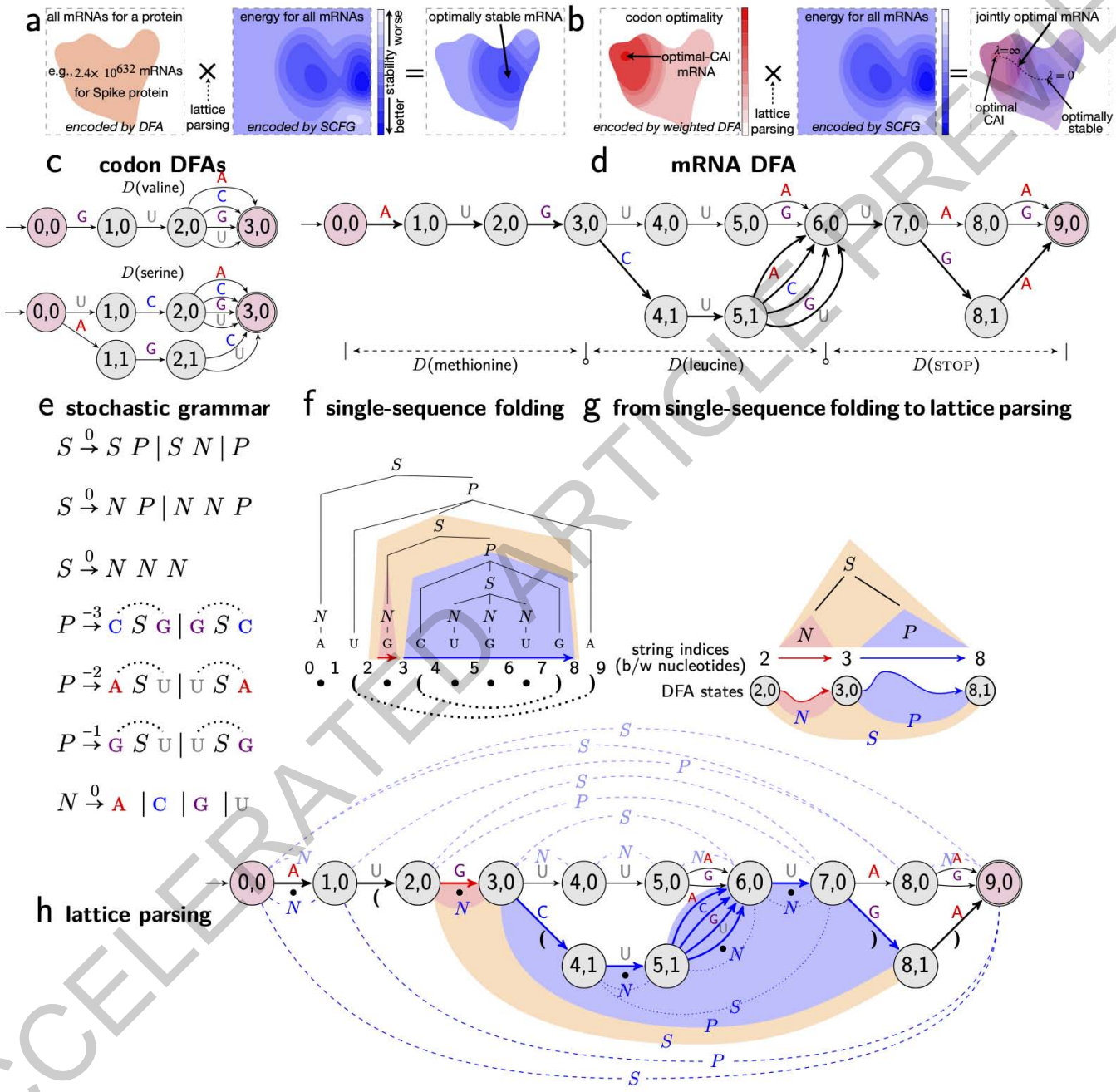




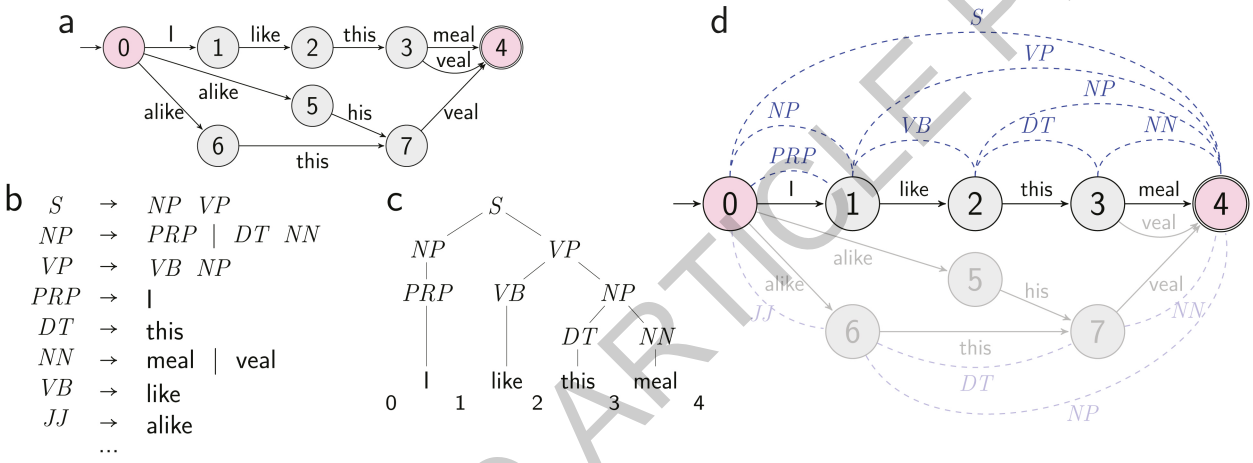








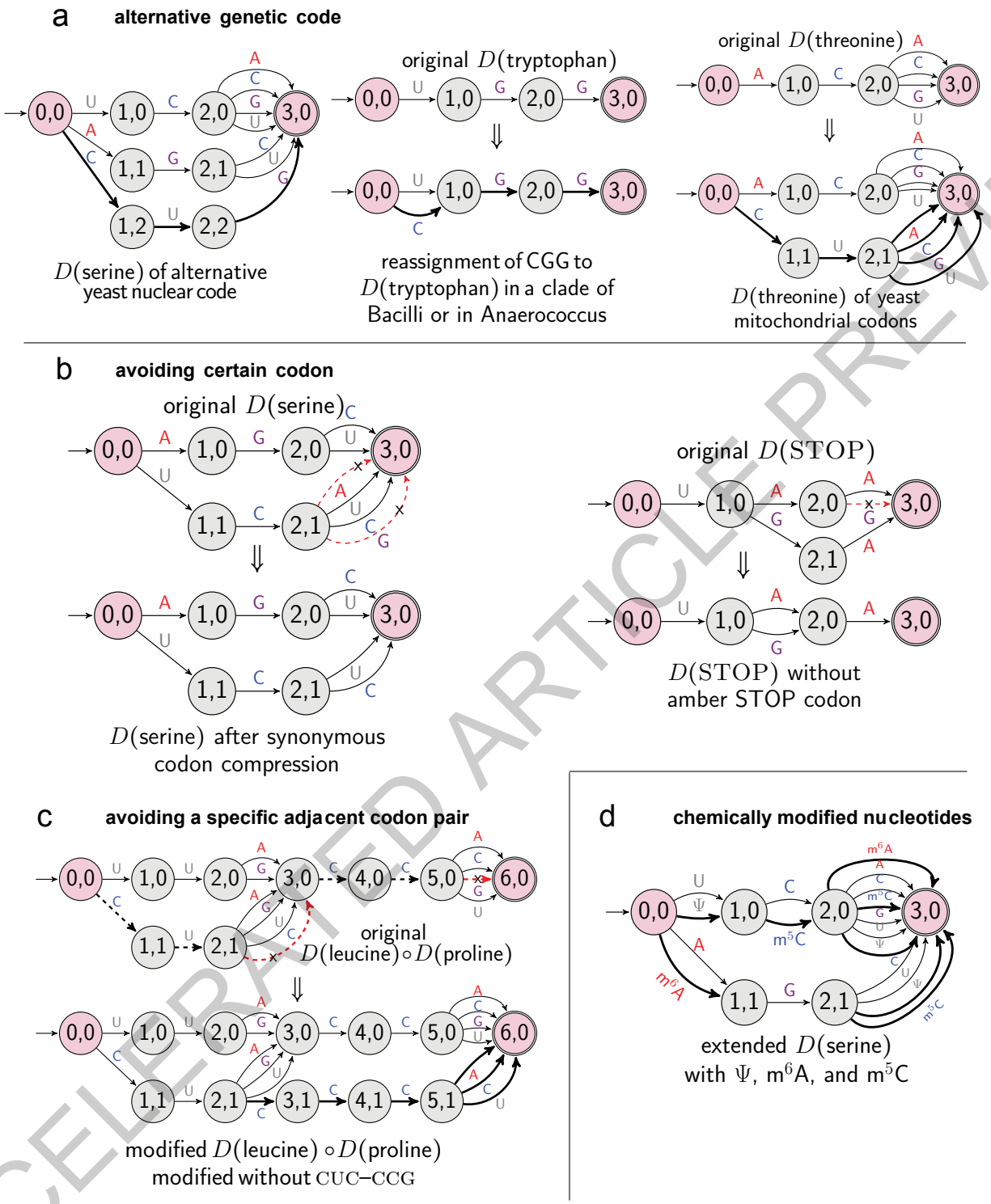
Extended Data Fig. 1



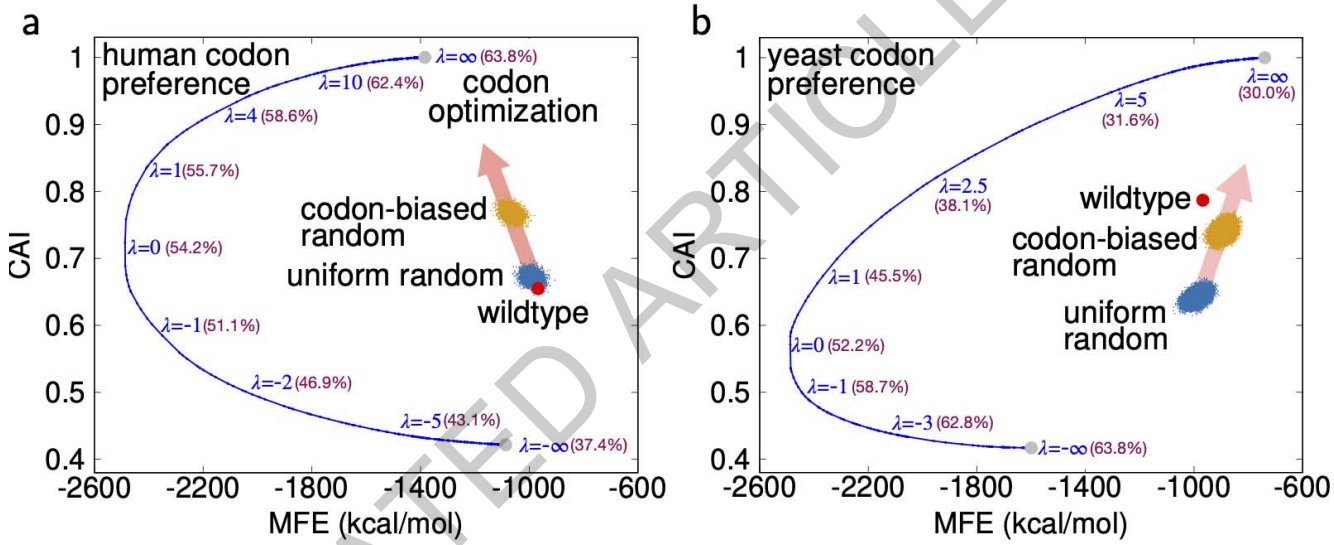
e

well-known connections in the literature:		
single-sequence parsing	⇔	single-sequence RNA folding
CKY ~1965, $O(n^3)$	⇔	Nussinov/Zuker ~1980, $O(n^3)$
new connections in this work:		
lattice parsing	⇔	mRNA design
Bar-Hillel *et al*, 1961, $O(\|D\|^3)$	⇔	our work, $O(n^3)$

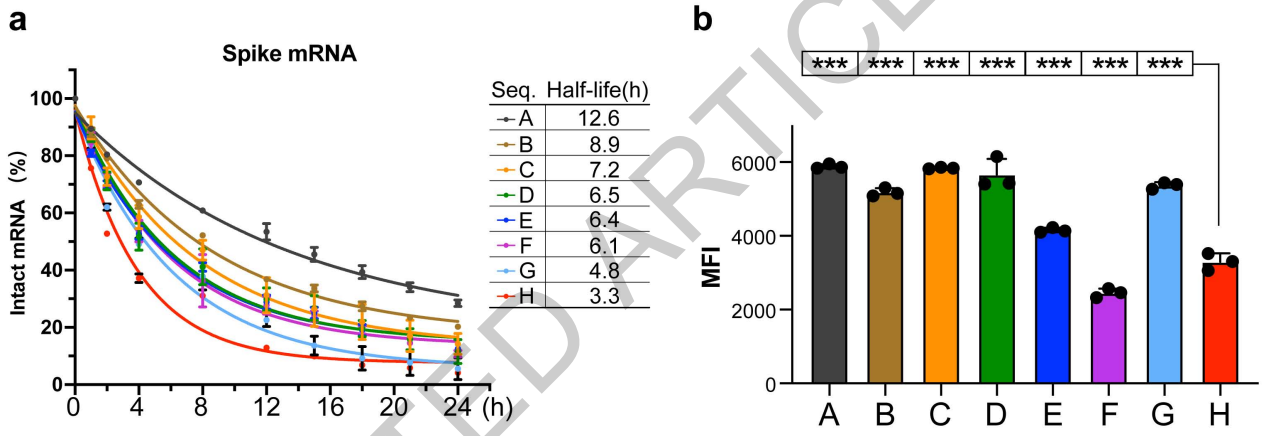
Extended Data Fig. 2



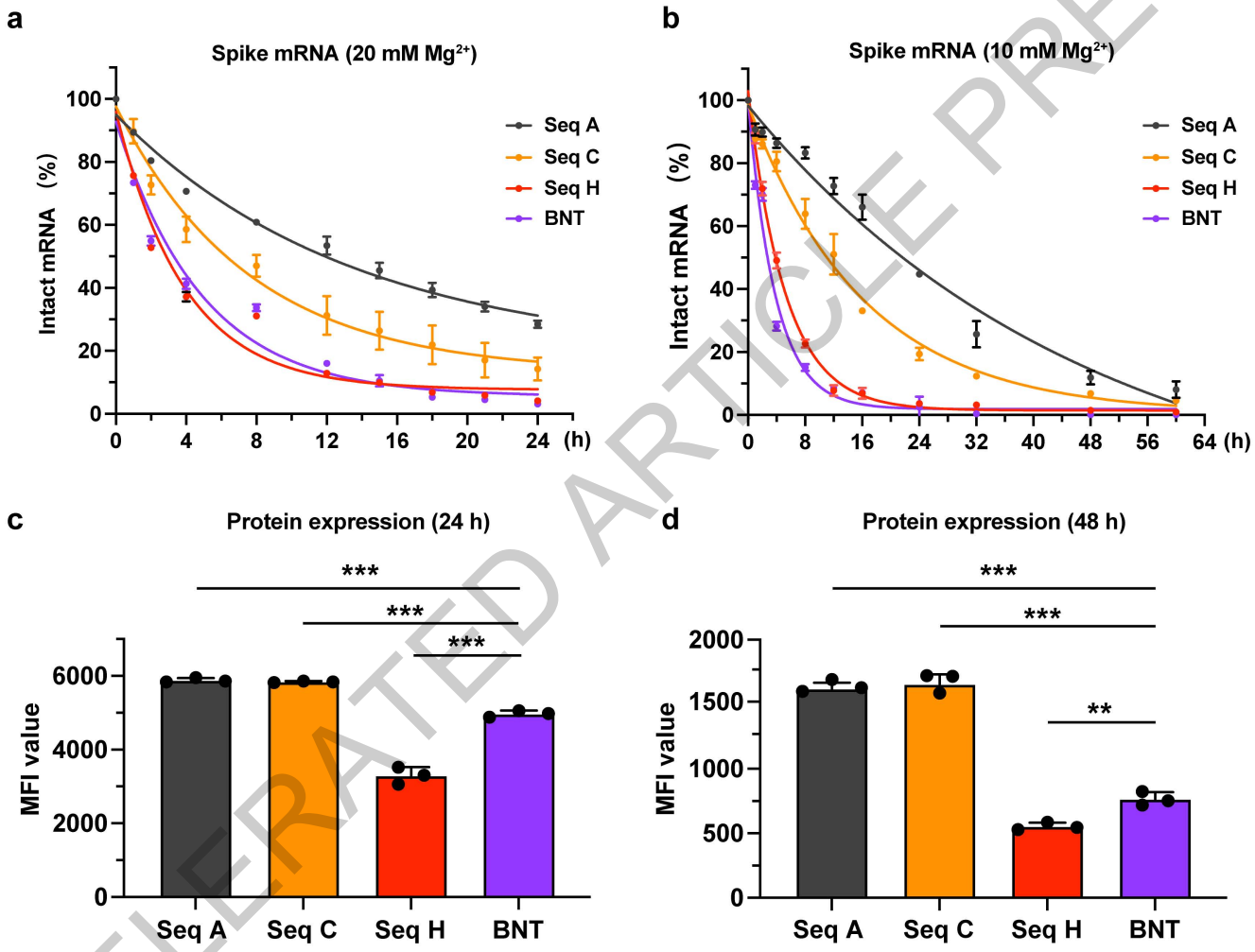
Extended Data Fig. 3



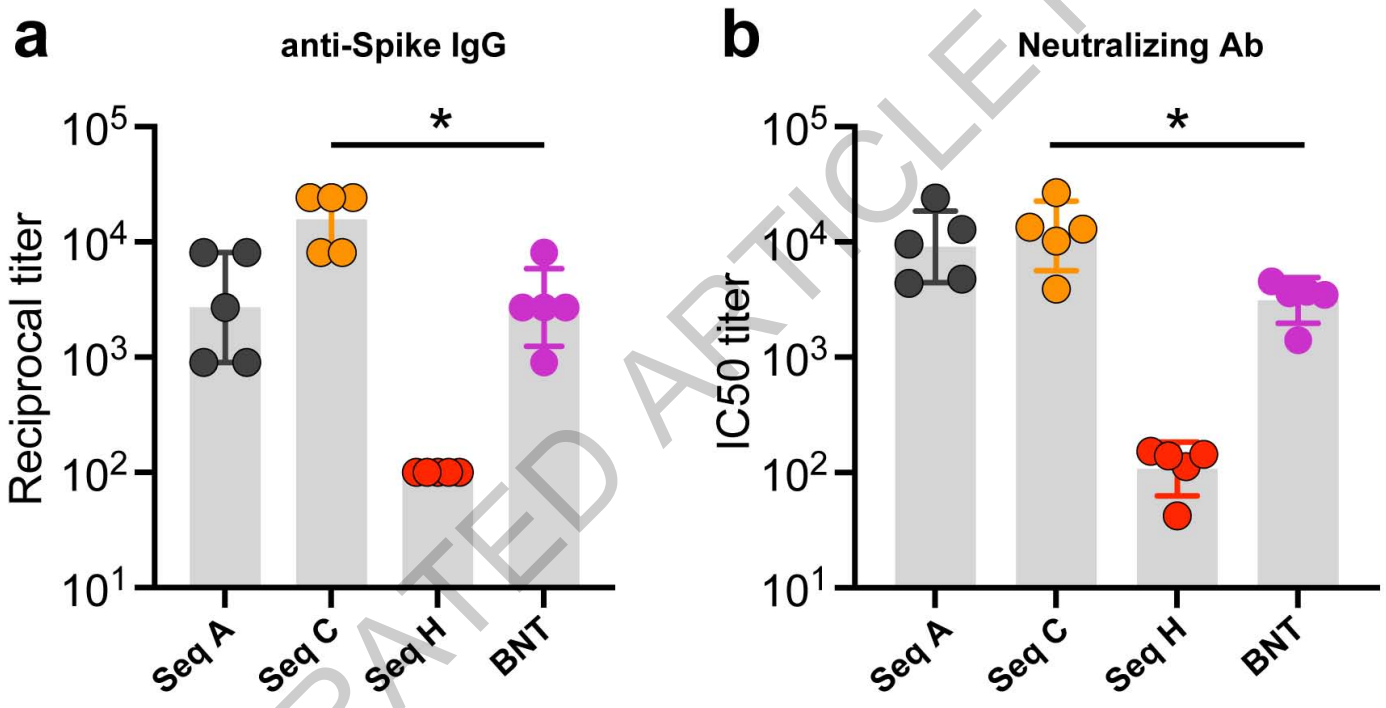
Extended Data Fig. 4



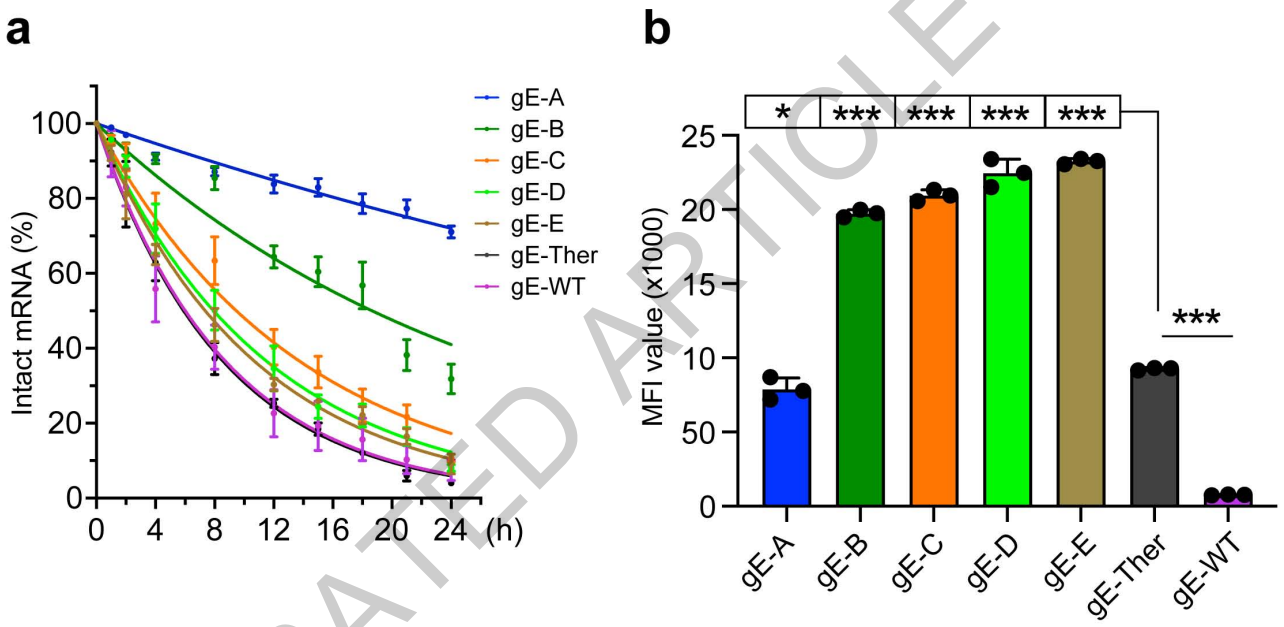
Extended Data Fig. 5



Extended Data Fig. 6



Extended Data Fig. 7



Extended Data Fig. 8

		UTRs																			
sequence	MFE	StemiRNA COVID-19				BioNTech				Moderna				CureVac				human β-globin			
of CDS	of CDS	MFE	tot.	5'	3'	MFE	tot.	5'	3'	MFE	tot.	5'	3'	MFE	tot.	5'	3'	MFE	tot.	5'	3'
	kcal/mol					*kcal/mol*				*kcal/mol*				*kcal/mol*				*kcal/mol*			
A	-2,287.3	-2,328.9	11	9	2	-2378.2	9	5	4	-2,334.1	11	10	1	-2,314.6	1	0	1	-2,328.3	8	0	8
B	-2,213.2	-2,252.8	12	10	2	-2302.2	6	5	1	-2,258.9	13	12	1	-2,236.9	1	0	1	-2,251.1	29	0	29
C	-2,206.0	-2,243.5	11	9	2	-2294.3	7	6	1	-2,248.9	9	8	1	-2,230.6	1	0	1	-2,245.2	10	7	3
D	-1,967.4	-2,004.6	13	7	6	-2057.1	10	5	5	-2,011.5	11	10	1	-1,992.6	11	11	0	-2,005.7	8	0	8
E	-1,961.3	-2,003.1	13	11	2	-2057.5	16	5	11	-2,009.7	15	14	1	-1,989.9	4	3	1	-2,002.1	8	0	8
F	-1,969.3	-2,009.9	11	9	2	-2061.1	12	5	7	-2,012.8	11	10	1	-1,995.3	9	9	0	-2,009.1	19	0	19
G	-1,639.3	-1,680.4	34	7	27	-1742.1	62	4	58	-1,688.5	62	0	62	-1,674.1	25	25	0	-1,688.1	23	0	23
H	-1,244.4	-1,287.6	31	16	15	-1346.3	66	8	58	-1,292.9	18	17	1	-1,285.6	77	52	25	-1,286.4	21	0	21
CureVac	-1384.4	-1,423.0	14	8	6	-1478.5	64	5	59	-1,432.3	65	19	46	-1,419.1	77	61	16	-1,425.7	26	0	26
Moderna	-1,369.2	-1,411.1	41	7	34	-1464.3	66	6	60	-1,422.2	60	12	48	-1,406.3	53	45	8	-1,414.0	27	0	27
BioNTech	-1,217.2	-1,265.4	34	5	29	-1316.1	98	6	92	-1,269.2	46	15	31	-1,253.7	58	54	4	-1,266.8	23	0	23
MFE-opt.	-2,486.7	-2,523.7	1	1	0	-2574.8	10	5	5	-2,530.9	1	1	0	-2,512.8	3	3	0	-2,522.5	7	0	7
CAI-opt.	-1,384.1	-1,421.7	34	11	23	-1478.2	59	0	59	-1,430.2	35	7	28	-1,420.4	53	53	0	-1,426.8	33	5	28
Wildtype	-966.7	-1011.7	18	16	2	-1060.6	33	21	12	-1,018.5	26	25	1	-999.9	68	45	23	-1,016.3	75	9	66

Extended Data Table. 1

		UTRs																			
sequence of CDS	MFE of CDS	StemiRNA VZV MFE	tot.	5'	3'	BioNTech MFE	tot.	5'	3'	Moderna MFE	tot.	5'	3'	CureVac MFE	tot.	5'	3'	human β-globin MFE	tot.	5'	3'
	kcal/mol					*kcal/mol*				*kcal/mol*				*kcal/mol*				*kcal/mol*			
gE-A	-1,145.6	-1,198.2	14	12	2	-1,239.0	8	5	3	-1,192.8	12	10	2	-1,183.3	5	5	0	-1,185.1	9	0	9
gE-B	-1,082.9	-1,134.8	15	13	2	-1,177.1	5	5	0	-1,126.5	14	12	2	-1,116.5	5	5	0	-1,123.0	29	0	29
gE-C	-932.3	-987.1	10	8	2	-1,026.4	6	6	0	-988.8	10	8	2	-966.8	17	17	0	-970.0	11	7	4
gE-D	-845.4	-910.1	13	9	4	-945.6	10	5	5	-909.3	12	10	2	-885.1	3	0	3	-892.3	14	0	14
gE-E	-805.0	-865.8	14	12	2	-907.8	15	5	10	-871.4	16	14	2	-843.8	19	12	7	-852.0	18	0	18
gE-Ther	-592.2	-662.2	11	9	2	-695.6	11	5	6	-649.8	6	0	6	-643.7	11	0	11	-641.5	9	0	9
gE-WT	-485.7	-546.9	23	5	18	-599.4	32	4	29	-544.9	61	0	61	-534.6	22	22	0	-529.8	46	11	35

Extended Data Table. 2

nature portfolio

Corresponding author(s): Liang Huang; Yujian Zhang; David H. Mathews; Hangwen Li

Last updated by author(s): Apr 6, 2023

Reporting Summary

Nature Portfolio wishes to improve the reproducibility of the work that we publish. This form provides structure for consistency and transparency in reporting. For further information on Nature Portfolio policies, see our Editorial Policies and the Editorial Policy Checklist.

Statistics

For all statistical analyses, confirm that the following items are present in the figure legend, table legend, main text, or Methods section.

n/a	Confirmed	
☐	☒	The exact sample size (*n*) for each experimental group/condition, given as a discrete number and unit of measurement
☐	☒	A statement on whether measurements were taken from distinct samples or whether the same sample was measured repeatedly
☐	☒	The statistical test(s) used AND whether they are one- or two-sided *Only common tests should be described solely by name; describe more complex techniques in the Methods section.*
☒	☐	A description of all covariates tested
☒	☐	A description of any assumptions or corrections, such as tests of normality and adjustment for multiple comparisons
☐	☒	A full description of the statistical parameters including central tendency (e.g. means) or other basic estimates (e.g. regression coefficient) AND variation (e.g. standard deviation) or associated estimates of uncertainty (e.g. confidence intervals)
☐	☒	For null hypothesis testing, the test statistic (e.g. *F*, *t*, *r*) with confidence intervals, effect sizes, degrees of freedom and *P* value noted *Give P values as exact values whenever suitable.*
☒	☐	For Bayesian analysis, information on the choice of priors and Markov chain Monte Carlo settings
☒	☐	For hierarchical and complex designs, identification of the appropriate level for tests and full reporting of outcomes
☒	☐	Estimates of effect sizes (e.g. Cohen's *d*, Pearson's *r*), indicating how they were calculated

Our web collection on statistics for biologists contains articles on many of the points above.

Software and code

Policy information about availability of computer code

Data collection	The LinearDesign source code is available to all parties on GitHub (https://github.com/LinearDesignSoftware/LinearDesign), and is free for academic and research use.
Data analysis	Clang (11.0.0) is used to compile LinearDesign source code. Vienna RNAfold from ViennaRNA package (version 2.4.14; open source) is used for predicting and drawing the secondary structure of mRNA sequence, and calculating the Minimum Free Energy (MFE) of secondary structures. For the wet lab experiments, GraphPad Prism 8.0 was used for the data analysis. Flow cytometry data were analyzed by FlowJo 10.1.

For manuscripts utilizing custom algorithms or software that are central to the research but not yet described in published literature, software must be made available to editors and reviewers. We strongly encourage code deposition in a community repository (e.g. GitHub). See the Nature Portfolio guidelines for submitting code & software for further information.

Data

Policy information about availability of data

All manuscripts must include a data availability statement. This statement should provide the following information, where applicable:

- Accession codes, unique identifiers, or web links for publicly available datasets
- A description of any restrictions on data availability
- For clinical datasets or third party data, please ensure that the statement adheres to our policy

The UniProt sequences used to estimate the time complexity of LinearDesign are included in Supplementary Tab. 1 and deposited at our figshare repository https://

doi.org/10.6084/m9.figshare.22193251. The COVID-19 and VZV mRNA coding region sequences and UTR sequences used in the biological experiments are included at the end of Supplementary Information file and available on our figshare repository. Source data of the animal experiments is provided with this paper, and all source data of wet lab experiments is available on that repository.

Human research participants

Policy information about studies involving human research participants and Sex and Gender in Research.

Reporting on sex and gender	N/A
Population characteristics	N/A
Recruitment	N/A
Ethics oversight	N/A

Note that full information on the approval of the study protocol must also be provided in the manuscript.

Field-specific reporting

Please select the one below that is the best fit for your research. If you are not sure, read the appropriate sections before making your selection.

☒ Life sciences ☐ Behavioural & social sciences ☐ Ecological, evolutionary & environmental sciences

For a reference copy of the document with all sections, see nature.com/documents/nr-reporting-summary-flat.pdf

Life sciences study design

All studies must disclose on these points even when the disclosure is negative.

Sample size	In the animal study, six mice (for COVID mRNA vaccine experiments) and five mice (for VZV mRNA vaccine experiments) were used in the corresponding experiments, respectively. The sample size of mice in each group was determined based on general animal study practice. Five or six mice per group were commonly used, which can also been seen in other publications (Nature 58, 567-571 (2020); Nat Commun 12, 2893 (2021); Molecular Therapy 29.6 (2021): 1970-1983.)
Data exclusions	There is no data exclusion in our study.
Replication	In vitro experiments were independently repeated in triplicate. All replication attempts were successful. Animal experiments were completed once. Gel electrophoresis experiments were repeated three times to obtain similar results.
Randomization	Animals were randomly allocated into each group. No specific randomization method was used. For other experiments, we performed side-by-side comparison at the same time to keep the experimental condition uniform. Therefore no randomization is needed.
Blinding	The investigators were not blinded to the data collection as all the assays were run by the same team that performed the animal immunization.

Reporting for specific materials, systems and methods

We require information from authors about some types of materials, experimental systems and methods used in many studies. Here, indicate whether each material, system or method listed is relevant to your study. If you are not sure if a list item applies to your research, read the appropriate section before selecting a response.

Materials & experimental systems

n/a	Involved in the study
☐	☒ Antibodies
☐	☒ Eukaryotic cell lines
☒	☐ Palaeontology and archaeology
☐	☒ Animals and other organisms
☒	☐ Clinical data
☒	☐ Dual use research of concern

Methods

n/a	Involved in the study
☒	☐ ChIP-seq
☐	☒ Flow cytometry
☒	☐ MRI-based neuroimaging

Antibodies

Antibodies used	anti-RBD Fc chimeric mAb (Cat: 40150-D001, Sino Biological) Clone #D001 PE-anti-human IgG Fc (Cat: 410707, Biolgend) Clone M1310G05 HRP-conjugated goat anti-mouse IgG Ab (Cat: 31430, Invitrogen) Polyclonal Anti-VZV gE protein antibody (Cat: 272686, Abcam) Clone #9 Goat Anti-Mouse IgG H&L (PE)) (Cat: 97024, Abcam) Polyclonal Goat Anti-Mouse IgG Fc (HRP) (Cat: 97265, Abcam) Polyclonal
Validation	anti-RBD Fc chimeric mAb: Du L, et al. (2009) The spike protein of SARS-CoV--a target for vaccine and therapeutic development. Nat Rev Microbiol. 7 (3): 226-36. Anti-VZV gE protein antibody: Wu S et al. Transcriptome Analysis Reveals the Role of Cellular Calcium Disorder in Varicella Zoster Virus-Induced Post-Herpetic Neuralgia. Front Mol Neurosci 14:665931 (2021).

Eukaryotic cell lines

Policy information about cell lines and Sex and Gender in Research

Cell line source(s)	HEK-293 cell line from ATCC (Cat# CRL-1573™) was used.
Authentication	Cell line was not authenticated.
Mycoplasma contamination	The cells were tested negative for mycoplasma contamination. MycoBlue Mycoplasma Detector (Vazyme) was used for detection.
Commonly misidentified lines (See ICLAC register)	No commonly misidentified cell lines were used.

Animals and other research organisms

Policy information about studies involving animals; ARRIVE guidelines recommended for reporting animal research, and Sex and Gender in Research

Laboratory animals	C57BL/6 mice (6-8 weeks, female) were used in this study. Mice were maintained on 12 h light:dark cycles with a housing temperature between 20–24 °C and 40-60% humidity.
Wild animals	The study did not involve wild animals.
Reporting on sex	Only female mice were used in this study without specific consideration of the sex impact on the results. Though publications have shown that male and female mice may differ in immune responses to vaccination (PNAS, 2018 Dec 4; 115(49): 12477–12482.). We followed a general practice using female mice in COVID-19 vaccine studies as used in other studies (Nature 586, 567–571 (2020); Cell 182, 1271–1283.e1–e7, September 3, 2020).
Field-collected samples	No field-collected samples were involved in this study.
Ethics oversight	All mice studies were performed in strict accordance with the guidelines set by the Chinese Regulations of Laboratory Animals and Laboratory Animal-Requirements of Environment and Housing Facilities. Animal experiments were carried out in compliance with the approval protocol from the Institutional Animal Care and Use Committee (IACUC) of Shanghai Model Organisms Center, Inc..

Note that full information on the approval of the study protocol must also be provided in the manuscript.

Flow Cytometry

Plots

Confirm that:

☒ The axis labels state the marker and fluorochrome used (e.g. CD4-FITC).

☒ The axis scales are clearly visible. Include numbers along axes only for bottom left plot of group (a 'group' is an analysis of identical markers).

☒ All plots are contour plots with outliers or pseudocolor plots.

☒ A numerical value for number of cells or percentage (with statistics) is provided.

Methodology

Sample preparation	Human embryonic kidney 293 cells (HEK293) (ATCC) were cultured in Dulbecco's Modified Eagle's Medium (DMEM) (Hyclone) containing 10% fetal bovine serum (FBS) (GEMINI) and 1% Penicillin-Streptomycin (Gibco). All cells were cultured at 37 °C in a 5% CO2 condition. For the measurement of protein expression, cells were transfected with mRNA using Lipofectamine MessengerMAX (Thermo

	Scientific). Briefly, a mix of 2 μg mRNA and 6 μL of Lipofectamine reagent was prepared following the manual instructions and then incubated with cells for 24 or 48 hours. For flow cytometric analysis, cells were collected and stained with live/dead cell dye (Fixable Viability Stain 510, BD) for 5 min. After washing, cells were incubated with anti-RBD chimeric mAb (1:100 dilution, Sino Biological) for 30 min, followed by washing and incubation with PE-anti-human IgG Fc (1:100 dilution, Biolgend) for 30 min. Samples were analyzed on BD Canto II (BD Biosciences). Data were processed using FlowJo V10.1 (Tree Star).
Instrument	BD FACSCanto II (Serial # : R33896203261).
Software	Flowjo version 10.1 was used in FACS analysis.
Cell population abundance	After gating the singlet cells, a total of 10,000 cells were collected for each independent assay.
Gating strategy	In our FACS experiments, only homogeneous cells (HEK293) were used for the evaluation of specific protein translation. In this case, only one fluorescent staining was used to assess the intensity. No other unique gating strategy was applied except for the exclusion of doublets and dead cells.

☐ Tick this box to confirm that a figure exemplifying the gating strategy is provided in the Supplementary Information.