Machine Learning Driven Simulations of SARS-CoV-2 Fitness Landscape

Aleksander E. P. Durumeric,[*,†,‡] Jonas Köhler,[†] Katarina Elez,[†] Lluís Raich,[†] Patricia A. Suriana,[†,¶] and Terra Sztain[*,†,§]

†*Department of Mathematics and Computer Science, Freie Universität Berlin, Arnimallee 12, 14195 Berlin, Germany*

‡*Institute for Theoretical Physics, Heidelberg University, 69120 Heidelberg, Germany*

¶*Department of Computer Science, Stanford University, 353 Jane Stanford Way, Stanford, CA 94305, United States*

§*Department of Medicinal Chemistry, University of Michigan, 428 Church St, Ann Arbor, MI 48109, United States*

E-mail: aleksander.durumeric@fu-berlin.de; tsztain@med.umich.edu

Abstract

SARS-CoV-2 infection is mediated by interactions between the receptor binding domain (RBD) of viral spike proteins and host cell angiotensin converting enzyme 2 (ACE2) receptors. Mutations in the spike protein are the primary cause for neutralizing antibody escape leading to breakthrough infections. We characterize the fitness landscape underpinning future variants of concern by combining supervised machine-learning and Markov Chain Monte Carlo. Leveraging deep mutational scanning (DMS) data characterizing the binding affinity between RBD mutants to the ACE2 receptor, we predict variants of concern not seen in the training data and sample statistics of the fitness landscape. These simulations provide insight into the relationship between RBD

sequence elements and offer a new perspective on utilizing DMS to predict emerging viral strains.

Introduction

The SARS-CoV-2 virus leading to the COVID-19 pandemic continues to evolve, accumulating mutations which evade immune responses and therapies.[1–4] Preparing for future variants remains challenging due to the vast number of mutations that could give rise to virulent strains. Most problematic mutations occur in the receptor binding domain (RBD) of the spike protein, the first point of contact during infection.[5] Starr et al. performed deep mutational scanning (DMS) on the spike RBD in 2020,[6] providing sequence-function information characterizing the effect of 10^5 possible mutations on RBD angiotensin converting enzyme 2 (ACE2) receptor binding, a strong indicator of viral fitness. Additional experiments initialized from the B1.351 variant in 2021[7] and omicron BA.1 and BA.2 in 2022[8] were carried out with library generation methods optimized to produce saturating point mutations at each position, characterizing a smaller number of additional strains (10^4). However, these studies are inherently unable to provide comprehensive information across all possible mutations (*e.g.*, $20^{200} \approx 10^{260}$ for a 200 residue protein). Furthermore, the reported fitness estimate for any single sequence is noisy, impeding the isolation of important mutants and hampering understanding the effect of individual mutations.

The ability to computationally predict the effect of any mutation on viral fitness would allow scientists to quickly isolate which mutations are problematic. However, this dependence is difficult to model. Due to the complex nature of protein sequence-function relationships, the effect of combined mutations cannot be inferred from adding or averaging the effects of single mutations.[9–12] Machine learning (ML) has been transformative for molecular biology, with approaches such as Alphafold2 and ESMfold demonstrating that protein crystal structures can be predicted accurately from protein sequence.[13–17] ML techniques have similarly

established the ability to directly learn complex sequence-function relationships to guide protein design.[10,11,18–22] These approaches have since been used to create models relating mutations in the RBD to changes in various properties.[12,23–25]

Previous work has focused on characterizing the average accuracy of RBD fitness predictors (e.g., mean squared error) and designed small numbers of high-fitness sequences.[12,23–25] However, fitness predictors inherently contain information characterizing how all possible groups of mutations may affect the activity of a given sequence. Identifying groups of influential residues directly from a trained predictor would provide valuable information on evolving viral strains.

In this work we focus on characterizing collective information in fitness predictors that are trained to predict RBD binding affinity estimates given in DMS data. This is achieved through sampling the fitness landscapes implied by predictors using Markov Chain Monte Carlo (MCMC). We find that imperfect predictors can be refined for simulation using maximum entropy biases, resulting in a straightforward procedure for generating novel diverse sequences with strong predicted fitness; the observed statistics of these sequences may then be analyzed to understand the role of various mutations. We show that this simple procedure predicts several SARS-CoV2 mutations identified by the World Health Organization in variants of concern, including those present in omicron strains. We leverage the generative diversity of this procedure to report the propensity of mutations at every position in the RBD and analyze the epistatic relationship between mutants. Collectively, these findings demonstrate the importance of training predictors on multi-mutation data, the capacity of ML to characterize higher order mutations more accurately than linear models, and the utility of sampling fitness landscapes defined by predictors via MCMC. This strategy is expected to be applicable to other mutation prediction tasks and evolving pathogens.

Results and Discussion

Machine learning architectures predict RBD binding affinity

The quality of an explored fitness landscape inherently depends on the accuracy of the predictor generating it. Multiple predictors linking sequence to RBD binding affinity were trained via least-squares regression on DMS data. The DMS dataset used for training[6] was experimentally initialized at the wild type (WT) RBD sequence and contains 105,525 variants with corresponding relative K_{D} values. To create a more expressive target for fitting, values were transformed to correspond to the binding free energy via

$$\Delta G = -RT \ln(K_{\mathrm{D}}), \tag{1}$$

where ΔG denotes the change in free energy upon ACE2 binding, $R = 8.314 \times 10^{-3}$ kJ/mol, and $T = 298K$. As experimental K_{D} values are reported as in comparison to the WT sequence, regression was performed on $\Delta\Delta G = \Delta G - \Delta G_{\mathrm{WT}}$. Predictors included linear models, a fully connected neural network (multilayer perceptron, MLP),[26] a message passing graph neural network (MPGNN),[27,28] and gradient boosted decision trees[29,30] with and without dropout (DART);[31] results are shown in **Figure 1A-C**.

Among all searched models and hyperparameters (**Tables S2-S3**), an MLP using a one-hot sequence embedding with a single hidden layer achieved the lowest mean absolute error (MAE), outperforming both the MPGNN and DART with test MAEs of 0.29, 0.35, and 0.49 kJ/mol. Additional performance metrics, such as MSE, are found in the SI. These errors were derived from a randomly selected test set. Predictors were also compared to a physics based $\Delta\Delta G$ prediction using Rosetta, whose predicted binding scores did not correlate well with experimental $\Delta\Delta G$ (**Figure S1**). Validating the accuracy of variant effect predictors, such as our trained models, is a complex task,[32,33] as differences in intended use can change which aspects of estimated performance are relevant. However, it is not computationally

feasible to perform MCMC on all possible generated models. We find that MAE correlates strongly with other calculated regression metrics and here use it to select models for later surface generation; alternative setups are considered in later sections. Additional results on prediction accuracy are found in the SI.

A MPGNN with sufficiently expressive hyperparameters should ostensibly be able to outperform an MLP due to the addition of structural information. However, similar to previous studies,[19] we do not see a gain in performance when using graph-based architectures. The tested MPGNNs were created using only the WT structure, as representative structures are not known for most mutants, which may represent a possible avenue for improvement. However, as binding affinities are ensemble properties, structural information may not be helpful as an inductive bias to predict mutational $\Delta\Delta G$s in general.

Unsurprisingly, the MLP predicts the fitness of higher order mutations with far greater accuracy than simply adding or averaging the impact of point mutations, highlighting the importance of accounting for non-additive effects when analyzing DMS data.[9–12] The MAE of the MLP remains low and relatively consistent independent of the number of mutations, up to 9 combined mutations (**Figure 2**), the maximum number present in the training set. However, the number of data points decreases with higher mutation counts, and despite maintaining low MAE, the Spearman's rank correlation coefficient drops significantly with

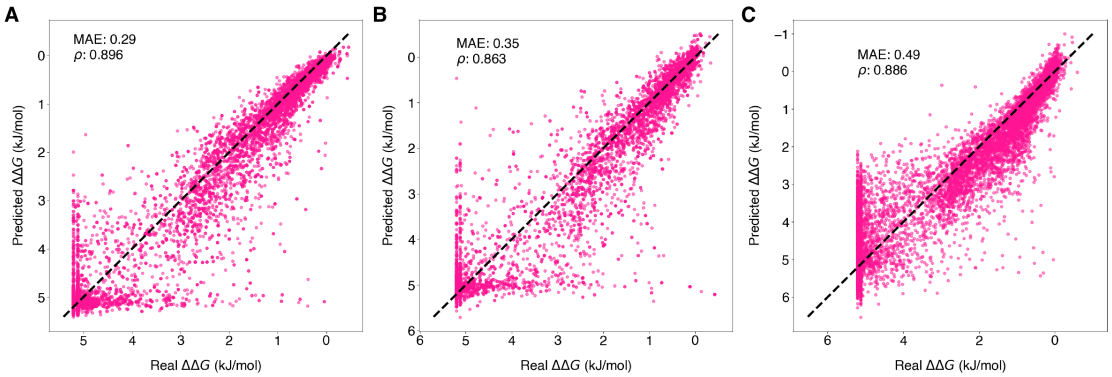

Figure 1: Summary of ML models used to predict ACE2 binding from RBD sequence. Predicted versus real $\Delta\Delta G$ values with MAE and Spearman's rank coefficients from A. MLP, B. MPGNN, and C. DART.

greater than six mutations from WT, likely since these mutants tend to all fall in the low binding, high noise region (**Figure S2**).

Adding classification layers to filter non-binding sequences

While convenient, it is unclear if MAE averaged over all test sequences is a faithful indicator of model quality for landscape generation. Possible reasons include dataset bias, as the majority of training examples are less stable binders than the WT, biasing training to focus on low binding stability sequences; in contrast, these low stability binders may be less important for defining the "basins" of a fitness landscape. This effect may be compounded by experimental detection limits, which may increase signal noise for sequences with lower binding stability.[34,35]

To avoid fitting to the intricacies of weak-binding mutants, modified training approaches were considered that reduced the impact of low binding affinity data. These setups were used

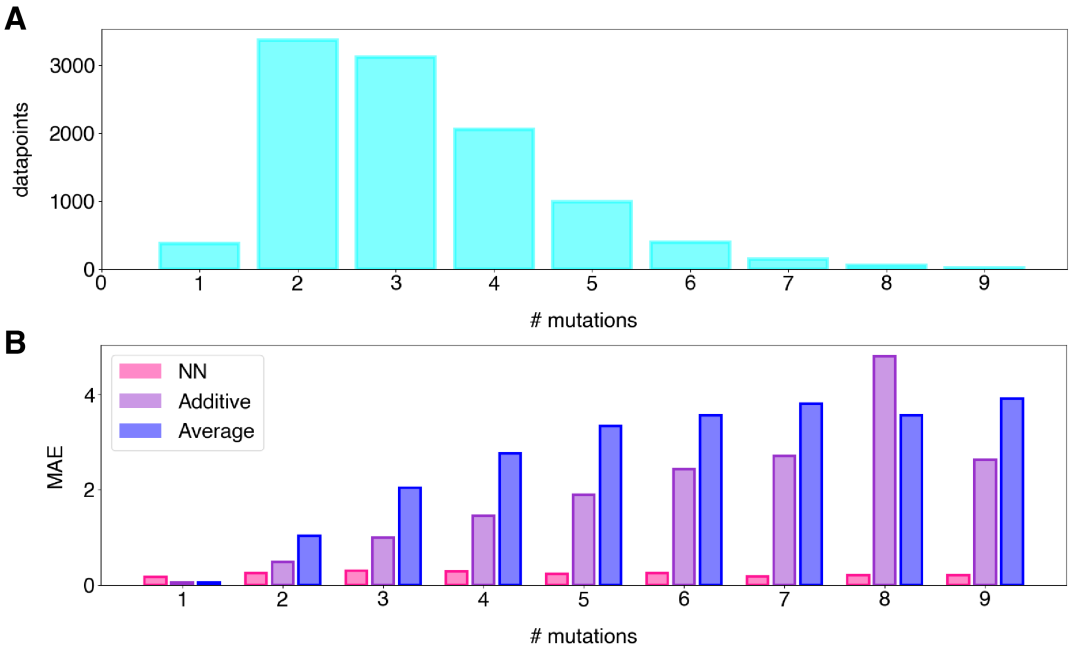

Figure 2: Comparison of mutation count. A. Number of data points containing each number of combined mutations in the test set. B. Average MAE using the neural network, additive effect of point mutations, or average effect of point mutations.

to retrain the MLP using the optimal hyperparameters found in previous sections. A threshold based on the experimental binding free energy was used to label a mutant as "binding" or "non-binding"; regression training then focused on minimizing prediction error on mutants in the "binding" category. Multitask and sequential approaches to training regression models that focused on the accuracy of binders were tested (SI) and may represent promising avenues towards improving predictions of noisy high throughput experimental data, especially in cases where measurements are only calculated for binders above a threshold, such as drug screening campaigns where properties are only measured for "hits".[34] Despite improved prediction performance on binders, these classification-augmented models did not produce significantly different fitness landscapes (SI).

Data from experiments initiated with variants

Several VOC out-competed the WT strain early in the pandemic.[36] As a result, subsequent RBD-ACE2 binding investigations were carried out with different starting sequences by various researchers. In 2021 Bloom and colleagues published additional libraries initiated from N501Y, E484K, B1.351 (N501Y, E484K, and K417N),[7] and in 2022 Bloom and colleagues published DMS omicron BA.1 and omicron BA.2 datasets.[8] These libraries followed the design of the original WT study but used new library construction techniques aimed at efficiently saturating point mutations. These datasets represent valuable sources of high-mutation data; for example, omicron represented an unprecedented evolutionary jump with over 30 spike mutations, including 15 in the RBD region.[37] However, as these alternative data sets contain an order of magnitude fewer examples ($\approx 10^4$) than the original DMS dataset (Table S4), they are used in conjunction with the WT data (SI). First, models pre-trained on the WT data set were fine tuned with BA.1 data; however, this did not improve performance. Simultaneously training on all data proved more effective, resulting in an reduced overall MAE; however, these models did not produce significantly different fitness surfaces (SI).

Characterizing the fitness landscape

Trained models that accurately predicted RBD binding affinity from sequence were used to explore fitness landscapes underpinning future VOCs using MCMC.[38] Results are shown for an MLP trained on the original DMS dataset (**Figure 1A**), as similar results were obtained when utilizing other best-in-class models, including those trained in a combined regression-classification setup and on extended datasets (SI).

Multiple definitions were used to create the fitness surface from binding affinity prediction. First, we propose that an informative landscape may be defined by assuming the system is characterized by a Boltzmann distribution

$$p(x) \propto f(x) := \mathrm{e}^{-\beta \Delta\Delta G(x)}, \tag{2}$$

where $p(x)$ is the probability of a particular sequence, $f(x)$ is proportional to this probability, β is an unknown positive scalar, and $\Delta\Delta G$ is the true change in binding affinity for a given mutant. We note that Eq. (2) corresponds to a maximum entropy distribution over sequences such that expectation of $\Delta\Delta G$ is fixed to a given value.[39,40]

This definition suggests performing MCMC using the predicted output of a trained model in place of $\Delta\Delta G$ with various values of β. However, this naïve approach produces sequences with saturating mutagenesis, such that every position containing a different amino acid compared to the WT (SI). We attribute this failure to limited model accuracy at extremely high mutation levels. Two modifications were made to produce more informative fitness landscapes. First, to avoid excessive mutations driving sampling outside the scope of the training data, sampled sequences were prohibited from exceeding 9 mutations relative to the the WT sequence. However, the resulting simulations still accumulated mutations up to the set maximum, with very little sampling below 6 mutations (SI). Therefore, to produce physically plausible mutation statistics and further focus sampling, we modified the sampled

distribution to be proportional to Eq. (3),

$$f^{\lambda}(x) := \mathbf{1}[\mathcal{N}(x) > \omega]\mathrm{e}^{-\beta g(x) - \lambda \mathcal{N}(x)} \tag{3}$$

where g corresponds to the predicted affinity, $\mathcal{N}$ is the number of residues which match those present in the WT sequence, λ controls bias towards the WT sequence, $\mathbf{1}$ denotes the 0-1 indicator function, and ω places a limit on the maximum number of mutations allowed during the simulation. λ may be viewed as a tuning parameter controlling the average number of mutations found during the simulation. Eq. (3) is sampled by increasing the probability of proposing the WT amino acid at each position, while maintaining an equal probability of proposing non-WT amino acids, akin to the "favor native" function in Rosetta (Figure S4); this modified proposal mechanism controls the stationary distribution of the Markov chain (SI).

Eq. (3) corresponds to a minimally adjusted form of $\mathrm{e}^{-\beta g(x)}$ modified to obtain a given average mutation count. Similar to Eq. (2), corrections of this form may be rigorously derived using maximum entropy arguments[39,40] and are used to incorporate experimental information into molecular dynamics force-fields.[41–43] As where corrections are often applied to modify simulations to match known experimental observations, we here use this approach to flexibly lower mutation propensity to account for an imperfect predictor.

UMAP projection of sequences sampled from f^{λ} shows several clusters which overlap with various VOC and unique clusters representing unseen regions of sequence space (Figure 3). Though none of the sequences have enough mutations to match omicron exactly due to imposed limits on mutation count, they are similar enough to overlap in the UMAP space. Visualizing the graph colored based on predicted binding affinity shows that each cluster contains a similar distribution of values (Figure S5).

Encouragingly, the model consistently predicts mutations found in the VOC (Table 1) without explicitly being trained on them, including the highly prevalent E484K and N501Y

mutations. These mutants are both present in alpha, beta, gamma, and mu strains.[44] E484K is also present in iota and zeta. The N501Y mutation remains present in all the omicron clades to date, whereas the E484 residue is instead mutated to an alanine in omicron. These mutations were consistently predicted in the simulations (Figure 4A,C), despite neither being present as a point mutation, or double mutation in the training dataset. Only higher order mutations containing these modified residues were present, yet the model was still able to accurately predict their role in increasing RBD binding affinity. N501 was also frequently mutated to a phenylalanine, a close relative of tyrosine. Though several mutants present in the BA.1 omicron variant were predicted by our simulation model trained in the absence of omicron-DMS data (G339G, N440K, S477N, E484K, N501Y), this current generation procedure is limited to sequences with up to 9 mutations away from the WT and accuracy of this trained models to higher levels of mutations is difficult to validate.

The mutation S477N, present in iota and omicron,[44] was also predicted by the simulations. Residue L452 is found to be mutated to R in delta, epsilon, kappa, and several

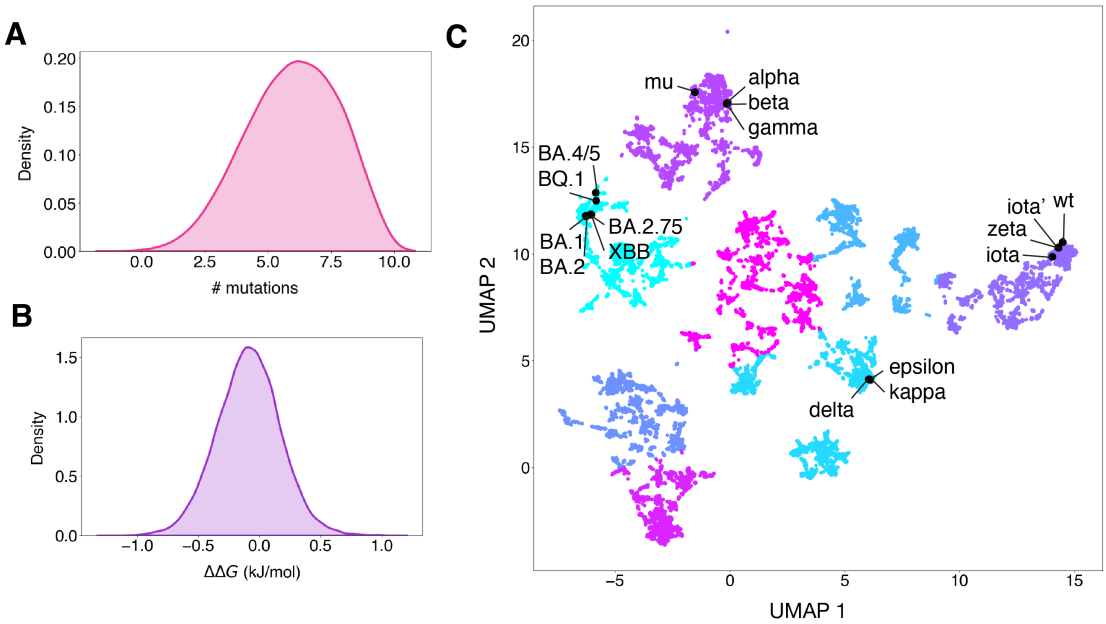

Figure 3: Monte Carlo simulations of MLP. A. Histogram showing distribution of number of mutations from the WT in accepted sequences. B. Histogram showing distribution of predicted $\Delta\Delta G$ in accepted sequences. C. UMAP projection of one-hot encoded sequences, colored based on k-means clustering. Variants of concern are labeled.

omicron variants, including BA.4/5, BQ.1, and CH.1.1. In our simulations, this residue was frequently mutated to a K, which also results in a non-polar to basic mutation and likely has similar effects on the structural properties. Other variants persisted in the simulation, which also emerged with omicron, including G339D, N440K, and N460K. Residues which have not been found in the VOC, but were predicted by the simulations include I358F, Y365W, V367F, Y453F, Q493V, Q498Y, Y505W, L518E, T523F, and V524F (Figure 4A,E). The mutants I358F and Y365W were found to enhance expression.[45] Additionally, V367F was found to be present with greater than 50% prevalence in three defined lineages: A.23, A.23.1, B.1.1.384, and Y453F was found in 15 lineages: DN.1.1.1, GD.1, FU.4, B.1.638, and 10 XBB descendants.[46,47]

The sequence frequency profiling provides an overview of the mutations independently. The correlation of mutation presence is another valuable metric for understanding epistatic relationships and predicting the effects of viral evolution. ML is useful in this case due to accurate predictions on the effect of combined mutations (Figure 2A-B). Through measuring co-occurrences of mutations during the simulation, we identified both positive and negative correlations between various mutations in the RBD (Figure 4B,D); additional co-mutational

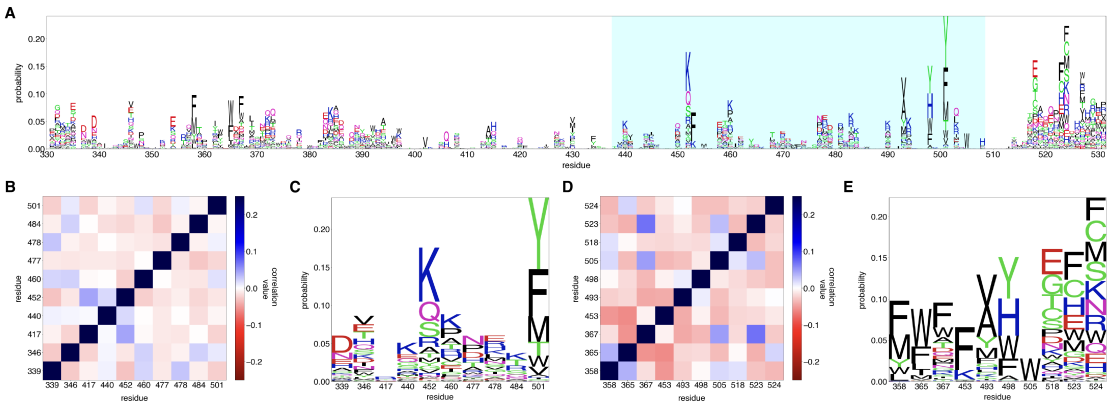

Figure 4: Analysis of simulated sequences. A. Logo plot of mutations accumulated during simulation. The ACE2 binding motif residues 438-508 are highlighted in cyan. B. Co-occurrence of known mutations predicted during simulation. C. Zoom into logo plot of known mutations predicted during simulation. D. Co-occurrence of novel mutations predicted during simulation. E. Zoom into logo plot of novel mutations predicted during simulation.

analysis and discussion is found in the SI.

Conclusion

We have demonstrated that a combination of supervised ML with DMS and MCMC is a viable approach for simulating fitness landscapes and extracting meaningful information on possible mutations. Using a multilayer perceptron we achieved highly accurate predictions of RBD binding to ACE2. This predictor was shown to approximate an informative fitness landscape when combined with a maximum-entropy bias, predicting mutations present in VOC (including omicron clades) without explicitly training on them. The valuable information from the larger libraries was central to our accuracy and is necessary for predicting higher order mutations. One limitation of our evolutionary simulations is the reduction of viral fitness to the single metric of ACE2 binding, however this has been proposed as one of the primary determinants of transmissibility in SARS-CoV-2.[48–50] Though only applied here to the case of SARS-CoV-2, DMS can provide a rich source of data on protein sequence-function relationships, and these methods are likely to be applicable to a broad range of topics within *in silico* evolution, including viruses, bacteria, cancer-related proteins, and directed evolution efforts towards protein engineering.

Methods

Data preparation

The WT ACE2 – RBD binding DMS datasets was obtained from `https://media.githubusercontent.com/media/jbloomlab/SARS-CoV-2-RBD_DMS/master/results/binding_Kds/binding_Kds.csv`. Data without K_{D} values were removed. Sequences associated with multiple entries within and across libraries were replace with their average value, resulting in 105,525 unique

sequences with K_{D} values. K_{D} values were transformed into $\Delta\Delta G$ values for regression. Data was randomly split into 80%/10%/10% points for train, valid, and test, respectively; note that for training of DART models, 5-fold cross validation was performed over the combination of these train and validation sets.

Alternative data sources

Additional datasets were preprocessed in the same manner as primary data; all universal resource locators are provided in Table S4. Only the MLP with optimal hyperparameters determined from the primary DMS data was used for prediction on the alternative data sets. When fine tuning the MLP on the alternative data sets, network weights after 25 epochs of training on WT data were used as starting weights for subsequent training on the alternative data set. The input layer weights were then frozen and training was performed on the hidden and output layer for 100 epochs.

DMS datasets often vary in terms of global scaling factors due to differences in experimental setup.[51] To account for this, sequences from different datasets were scaled using a learnable affine function. Each dataset was given an index, and those not originating from the WT dataset were passed through a affine transformation $f(x) = mx + b$ with learnable parameters m and b for each dataset. Doing so reduced the corresponding MSE (SI).

Prediction models

Baseline Models

As a baseline, the $\Delta\Delta G$ for individual point mutations were filtered and the test set was evaluated based on either adding or averaging the point mutant values. Sequences in the test set containing mutations without corresponding point mutation values were not included in the analysis in Figure 2.

Multilayer perceptron

A multilayer perceptron (MLP) implemented with the PyTorch[52] library was trained on various sequence embeddings to predict $\Delta\Delta G$. The optimal architecture used flattened one-hot embedding input layer, one hidden layer with 128 input features and 64 output features, activated with ReLU,[53] and a single value output layer; this was found via a hyperparameter search varying the number and size of hidden layers and input features (Table S1). Networks were created with ReLU[53] activation on all layers except the output layer, which used a linear activation. The mean squared error (MSE) between predicted and ground truth $\Delta\Delta G$ values was used as the loss function for training. A batch size of 128 and a learning rate of 10^{-4} with the Adam optimizer[54] was used to train until the convergence was observed via an uptick in the validation loss.

Sequence embedding Multiple different sequence embeddings (i.e., featurizations) were used as input to the MLP. A one-hot embedding was utilized based on the 20 canonical amino acids. Alternatively, a glycan one-hot embedding was defined by adding an additional token to the one-hot embedding denoting glycosylated asparagine residues based on the Nx[S/T] sequence motif.[55] A UniRep[56] embedding was used with the TAPEtokenizer[57] utilizing a 1900-dimensional hidden state layer. SeqVec embeddings generated via Bio Embeddings were additionally tested,[58] as were ProtTrans-Bert-BFD[59] and ESM1b[60] embeddings with default settings. Comparison of the performance associated with training on these various embeddings is shown in Table S2 when applied to the optimal MLP hyperparameters found for the one-hot embedding; one-hot embeddings performed best. Note that other modeling approaches used only the one-hot sequence embedding for testing, with the exception of DART models which utilized equivalent categorical variables based on residue type.

Message passing graph neural network

A message passing graph neural network was created using PyTorch to operate on a graph defined by a single structure of the WT RBD obtained from simulations performed by Barros et. al.[61] The structure was extracted from the trajectory by first clustering using cpptraj[62] via the hierarchical method with a random sieve of 10; this produced a total of 20 clusters. The average structure from the most populated cluster was then used to generate a residue graph with mdtraj.[63] A one-hot residue representation of each sequence was used to describe the nodes, while edges were defined based on the distance between alpha carbons of each residue in the structure. Edges were connected based on a window in sequence space of 10 residues in the N and C terminal directions, as well as a distance cutoff of 15 Å in 3D space. Distances were featurized into 10 dimensional feature vectors using a Gaussian expansion and scaled with a factor of 75. Messages were passed 7 times, each containing an MLP with a 128 feature hidden layer. Results of the hyperparameter search are shown in Table S3.

Multiple Additive Regression Trees with Dropout

Multiple Additive Regression Trees with Dropout (DART)[31] and gradient boosted decision tree models[29,30] were created using the Light Gradient Boosted Machine library.[64] For brevity, both gradient boosted decision trees (without dropout) and Multiple Additive Regression Trees with Dropout are referred to as DART. The optimal model was found to use 511 leaves, dropout, 1399 trees, a drop rate of 0.3, a maximum drop of 50, minimum leaf data size of 1, a categorical l2 of 0, and a categorical smoothing of 5. Hyperparameters were scanned over using multiple local grid searches, scanning over the the presence of dropout (*i.e.*, DART or gradient boosted decision trees), number of leaves, the rate of tree dropping, the presence of categorical smoothing and l2, the rate of tree dropping, and the minimum number of data allowed per leaf (SI).

All hyperparameter combinations were trained for 5000 updates to determine the optimal number of trees. We note that a relatively large number of leaves was found to be beneficial

to regression despite the low number of training points, likely implying a high degree of epistasis. Unlike other methods, hyperparameters were optimized using 5 fold cross validation performed on the the combined train and validation datasets used for other methods. All attempts treated inputs as categorical variables.

Combining classification and regression

The mean squared error (MSE) is represented by Eq. (4)

$$\mathcal{L}_1 = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \tag{4}$$

where n corresponds to the number of data points, y_i is the ground truth, and $\hat{y}_i$ the predicted value. For classification, the binary cross entropy (Eq. (5)) was used.

$$\mathcal{L}_2 = \frac{1}{n}\sum_{i=1}^{n} y_i \log(p_i) + (1 - y_i)\log(1 - p_i) \tag{5}$$

Here, p_i corresponds to the estimated probability of sample i being considered a binder vs. non-binder based on the cutoff shown by Eq. (6).

$$y_i = \begin{cases} 0 & \Delta\Delta G \leq \tau \\ 1 & \Delta\Delta G > \tau \end{cases} \tag{6}$$

τ is the threshold of the cutoff for being considered a binder. Various thresholds were tested and 1.82 kJ/mol was selected to maximize the amount of sequences classified as binders (SI).

These two loss functions were linearly combined to create the combined loss given by Eq. (7).

$$\mathcal{L} = \alpha\mathcal{L}_1^C + (1 - \alpha)\mathcal{L}_2 \tag{7}$$

where $\mathcal{L}_1^C$ denotes $\mathcal{L}_1$ only applied to binders as determined by the DMS binding value (Eq.

(8))

$$\mathcal{L}_1^C = \begin{cases} 0 & \Delta\Delta G \leq \tau \\ \mathcal{L}_1 & \Delta\Delta G > \tau \end{cases} \tag{8}$$

Various α were compared, with a final value of 0.005 chosen (SI).

Two alternative training setups trained an MLP using the combined loss in Eq. (8). First, a multitask approach trained a multi-head neural network to perform both classification and regression. All parameters were shared, except for a task-specific output layer: classification used a sigmoid output layer, while regression was performed with a linear output layer. Second, a sequential approach performed regression and classification using two separate networks; however, only sequences with both ground truth and predicted affinities above the threshold were fed to the regression network during training. Additional details are found in the SI.

Monte Carlo Simulations

Markov Chain Monte Carlo (MCMC) simulations were performed to explore fitness landscapes implied by the output of trained models. For MCMC simulation a random position between 1 and 201 was selected for mutation to one of 20 canonical amino acids. The proposed mutant was evaluated by the model and accepted or rejected according to a modified form of the Metropolis criteria.

MCMC performed using our trained model in place of $\Delta\Delta G$ in Eq. (2) leads to problematically high levels of mutations, likely driven by limited prediction accuracy on mutants far from those present in training set (SI). MCMC simulations are instead performed to target the following native-biased probability mass function

$$p^\lambda(x) \propto f^\lambda(x) := \mathbf{1}[\mathcal{N}(x) > \omega]\mathrm{e}^{-\beta g(x) - \lambda \mathcal{N}(x)} \tag{9}$$

where x denotes a amino acid sequence, $\mathcal{N}$ denotes the number of residues in x which

match those present in the WT sequence, $\lambda \in (-\infty, \infty)$ determines a bias towards the WT sequence, $\mathbf{1}$ denotes the 0-1 indicator function, and ω places a limit on the maximum number of mutations allowed during the simulation. ω was set such that mutants with greater than 9 mutations were not considered. Several values for β were scanned and 10 was chosen to keep the center of the distribution of $\Delta\Delta G$ outputs near 0, the fitness value of the WT. Values for λ were also tested and -5.15 was chosen to balance the diversity of the mutation landscape (Figure S4).

Eq. (9) is targeted by proposing single-residue mutations and accepting sequences using a modified form of the Metropolis procedure. Candidate 1-mutations were proposed using

$$q(x'|x) := q(n, r|x) \begin{cases} \frac{C}{L}, & r = \mathrm{WT}_n \\ \frac{1-C}{19L}, & r \neq \mathrm{WT}_n \end{cases}, \tag{10}$$

where n determines the index of the residue to mutate, r determines which amino acid the mutation will result in, and L denotes the number of mutable residues. Note that this a joint probability mass function over n and r; however, n does not appear as the corresponding marginal is uniform. Furthermore, note that this proposal distribution is asymmetric.

Proposed mutations are accepted according to a Metropolis-like criterion given by the following probability:

$$\alpha(x'|x) := \min\left[1, \frac{f(x')}{f(x)}\right]. \tag{11}$$

As shown in the SI, application of Eqs. (10) and (11) result in the stationary distribution given by Eq. (9) when defining λ to be $-\ln \frac{19C}{1-C}$. The selected value for λ corresponds to $C = 0.9$. In practice, sampling is performed by drawing a variate from a uniform distribution between 0 and 1 and comparing it to Eq. (11). If accepted, the mutant is set to be the current state of the MCMC simulation; if rejected, the state of the simulation is not changed. MCMC simulations were observed to return equivalent results on repeated runs and exhibited multiple crossing between groups of mutants, implying that simulations are well-converged.

Data availability

All code necessary for reproducing the manuscript can be found at `https://github.com/tsztain/ML_evolution_SARS-CoV-2` to be released at time of publication.

Acknowledgement

We gratefully acknowledge funding from the European Commission (Grant No. ERC CoG 772230 "ScaleCell"),the Deutsche Forschungsgemeinschaft DFG (RTG2433 DAEDALAUs, SFB1114/C03, FOR2518, NO825/4). LR acknowledges funding through European Union's Horizon 2020 research innovation programme under the Marie Sklodowska-Curie grant agreement No 897414, and AD acknowledges funding through a Flagship Initiative Engineering Molecular Systems Postdoctoral Fellowship.

TS conceived of the project. Data curation was carried out by TS and KE. Code for ML and analysis was developed by all authors, MCMC and analysis were performed by TS and AD. TS and AD wrote manuscript with contributions from all authors. We thank Frank Noé for invaluable insights and discussions throughout this project. We also appreciate the technical assistance of Cornelius Hoffmann.

Supporting Information Available

Supplementary materials will be made available upon publication.

- Figure S1. Physics based binding predictions.
- Figure S2. MLP predicted versus real $\Delta\Delta G$ by number of mutations from WT.
- Figure S3. Results from incorporating data from DMS initiated from variants.
- Figure S4. MCMC simulation parameters for tuning mutation accumulation.

- Figure S5. UMAP projection of one-hot encoded sequences, colored based on predicted $\Delta\Delta G$.
- Table S1. MLP hidden layer hyperparameter optimization
- Table S2. MLP embedding hyperparameter optimization
- Table S3. GMPNN hyperparameter optimization
- Table S4. Summary of DMS datasets

References

(1) Harvey, W. T.; Carabelli, A. M.; Jackson, B.; Gupta, R. K.; Thomson, E. C.; Harrison, E. M.; Ludden, C.; Reeve, R.; Rambaut, A.; Peacock, S. J.; others SARS-CoV-2 variants, spike mutations and immune escape. *Nature Reviews Microbiology* **2021**, *19*, 409–424.

(2) Krause, P. R.; Fleming, T. R.; Longini, I. M.; Peto, R.; Briand, S.; Heymann, D. L.; Beral, V.; Snape, M. D.; Rees, H.; Ropero, A.-M.; others SARS-CoV-2 variants and vaccines. *New England Journal of Medicine* **2021**, *385*, 179–186.

(3) Miller, J.; Hachmann, N. P.; Collier, A.-r. Y.; Lasrado, N.; Mazurek, C. R.; Patio, R. C.; Powers, O.; Surve, N.; Theiler, J.; Korber, B.; others Substantial neutralization escape by SARS-CoV-2 Omicron variants BQ. 1.1 and XBB. 1. *New England Journal of Medicine* **2023**, *388*, 662–664.

(4) Markov, P. V.; Ghafari, M.; Beer, M.; Lythgoe, K.; Simmonds, P.; Stilianakis, N. I.; Katzourakis, A. The evolution of SARS-CoV-2. *Nature Reviews Microbiology* **2023**, *21*, 361–379.

(5) Piccoli, L.; Park, Y.-J.; Tortorici, M. A.; Czudnochowski, N.; Walls, A. C.; Beltramello, M.; Silacci-Fregni, C.; Pinto, D.; Rosen, L. E.; Bowen, J. E.; others Mapping

neutralizing and immunodominant sites on the SARS-CoV-2 spike receptor-binding domain by structure-guided high-resolution serology. *Cell* **2020**, *183*, 1024–1042.

(6) Starr, T. N.; Greaney, A. J.; Hilton, S. K.; Ellis, D.; Crawford, K. H.; Dingens, A. S.; Navarro, M. J.; Bowen, J. E.; Tortorici, M. A.; Walls, A. C.; others Deep mutational scanning of SARS-CoV-2 receptor binding domain reveals constraints on folding and ACE2 binding. *cell* **2020**, *182*, 1295–1310.

(7) Starr, T. N.; Greaney, A. J.; Hannon, W. W.; Loes, A. N.; Hauser, K.; Dillen, J. R.; Ferri, E.; Farrell, A. G.; Dadonaite, B.; McCallum, M.; others Shifting mutational constraints in the SARS-CoV-2 receptor-binding domain during viral evolution. *Science* **2022**, *377*, 420–424.

(8) Starr, T. N.; Greaney, A. J.; Stewart, C. M.; Walls, A. C.; Hannon, W. W.; Veesler, D.; Bloom, J. D. Deep mutational scans for ACE2 binding, RBD expression, and antibody escape in the SARS-CoV-2 Omicron BA. 1 and BA. 2 receptor-binding domains. *PLoS pathogens* **2022**, *18*, e1010951.

(9) Starr, T. N.; Thornton, J. W. Epistasis in protein evolution. *Protein science* **2016**, *25*, 1204–1218.

(10) Yang, K. K.; Wu, Z.; Arnold, F. H. Machine-learning-guided directed evolution for protein engineering. *Nature methods* **2019**, *16*, 687–694.

(11) Freschlin, C. R.; Fahlberg, S. A.; Romero, P. A. Machine learning to navigate fitness landscapes for protein engineering. *Current opinion in biotechnology* **2022**, *75*, 102713.

(12) Taft, J. M.; Weber, C. R.; Gao, B.; Ehling, R. A.; Han, J.; Frei, L.; Metcalfe, S. W.; Overath, M. D.; Yermanos, A.; Kelton, W.; others Deep mutational learning predicts ACE2 binding and antibody escape to combinatorial mutations in the SARS-CoV-2 receptor-binding domain. *Cell* **2022**, *185*, 4008–4022.

(13) Jumper, J.; Evans, R.; Pritzel, A.; Green, T.; Figurnov, M.; Ronneberger, O.; Tunyasuvunakool, K.; Bates, R.; Žídek, A.; Potapenko, A.; others Highly accurate protein structure prediction with AlphaFold. *nature* **2021**, *596*, 583–589.

(14) Lin, Z.; Akin, H.; Rao, R.; Hie, B.; Zhu, Z.; Lu, W.; Smetanin, N.; Verkuil, R.; Kabeli, O.; Shmueli, Y.; others Evolutionary-scale prediction of atomic-level protein structure with a language model. *Science* **2023**, *379*, 1123–1130.

(15) Krishna, R.; Wang, J.; Ahern, W.; Sturmfels, P.; Venkatesh, P.; Kalvet, I.; Lee, G. R.; Morey-Burrows, F. S.; Anishchenko, I.; Humphreys, I. R.; others Generalized biomolecular modeling and design with RoseTTAFold All-Atom. *Science* **2024**, *384*, eadl2528.

(16) Elofsson, A. Progress at protein structure prediction, as seen in CASP15. *Current Opinion in Structural Biology* **2023**, *80*, 102594.

(17) Jänes, J.; Beltrao, P. Deep learning for protein structure prediction and design—progress and applications. *Molecular Systems Biology* **2024**, *20*, 162–169.

(18) Romero, P. A.; Krause, A.; Arnold, F. H. Navigating the protein fitness landscape with Gaussian processes. *Proceedings of the National Academy of Sciences* **2013**, *110*, E193–E201.

(19) Gelman, S.; Fahlberg, S. A.; Heinzelman, P.; Romero, P. A.; Gitter, A. Neural networks to learn protein sequence–function relationships from deep mutational scanning data. *Proceedings of the National Academy of Sciences* **2021**, *118*, e2104878118.

(20) Kouba, P.; Kohout, P.; Haddadi, F.; Bushuiev, A.; Samusevich, R.; Sedlar, J.; Damborsky, J.; Pluskal, T.; Sivic, J.; Mazurenko, S. Machine learning-guided protein engineering. *ACS catalysis* **2023**, *13*, 13863–13895.

(21) Notin, P.; Rollins, N.; Gal, Y.; Sander, C.; Marks, D. Machine learning for functional protein design. *Nature biotechnology* **2024**, *42*, 216–228.

(22) Puszkarska, A. M.; Taddese, B.; Revell, J.; Davies, G.; Field, J.; Hornigold, D. C.; Buchanan, A.; Vaughan, T. J.; Colwell, L. J. Machine learning designs new GCGR/GLP-1R dual agonists with enhanced biological potency. *Nature Chemistry* **2024**, 1–9.

(23) Wang, B.; Gamazon, E. R. Modeling mutational effects on biochemical phenotypes using convolutional neural networks: application to SARS-CoV-2. *Iscience* **2022**, *25*.

(24) Zhang, R.; Ghosh, S.; Pal, R. Predicting binding affinities of emerging variants of SARS-CoV-2 using spike protein sequencing data: observations, caveats and recommendations. *Briefings in Bioinformatics* **2022**, *23*, bbac128.

(25) Wang, G.; Liu, X.; Wang, K.; Gao, Y.; Li, G.; Baptista-Hon, D. T.; Yang, X. H.; Xue, K.; Tai, W. H.; Jiang, Z.; others Deep-learning-enabled protein–protein interaction analysis for prediction of SARS-CoV-2 infectivity and variant evolution. *Nature Medicine* **2023**, *29*, 2007–2018.

(26) Bishop, C. M.; Bishop, H. *Deep learning: Foundations and concepts*; Springer Nature, 2023.

(27) Gilmer, J.; Schoenholz, S. S.; Riley, P. F.; Vinyals, O.; Dahl, G. E. Neural message passing for quantum chemistry. International conference on machine learning. 2017; pp 1263–1272.

(28) Bronstein, M. M.; Bruna, J.; Cohen, T.; Veličković, P. Geometric deep learning: Grids, groups, graphs, geodesics, and gauges. *arXiv preprint arXiv:2104.13478* **2021**,

(29) Friedman, J. H. Greedy function approximation: a gradient boosting machine. *Annals of statistics* **2001**, 1189–1232.

(30) Friedman, J. H. Stochastic gradient boosting. *Computational statistics & data analysis* **2002**, *38*, 367–378.

(31) Vinayak, R. K.; Gilad-Bachrach, R. Dart: Dropouts meet multiple additive regression trees. Artificial Intelligence and Statistics. 2015; pp 489–497.

(32) Michael, R.; Kæstel-Hansen, J.; Mørch Groth, P.; Bartels, S.; Salomon, J.; Tian, P.; Hatzakis, N. S.; Boomsma, W. A systematic analysis of regression models for protein engineering. *PLOS Computational Biology* **2024**, *20*, e1012061.

(33) Freschlin, C. R.; Fahlberg, S. A.; Heinzelman, P.; Romero, P. A. Neural network extrapolation to distant regions of the protein fitness landscape. *Nature Communications* **2024**, *15*, 6405.

(34) Malo, N.; Hanley, J. A.; Cerquozzi, S.; Pelletier, J.; Nadon, R. Statistical practice in high-throughput screening data analysis. *Nature biotechnology* **2006**, *24*, 167–175.

(35) Patrone, P. N.; DiSalvo, M.; Kearsley, A. J.; McFadden, G. B.; Cooksey, G. A. Reproducibility in cytometry: Signals analysis and its connection to uncertainty quantification. *Plos one* **2023**, *18*, e0295502.

(36) Organization, W. H. Updated working definitions and primary actions for SARSCOV2 variants. 2024; `https://www.who.int/publications/m/item/updated-working-definitions-and-primary-actions-for--sars-cov-2-variants`.

(37) Organization, W. H. Classification of omicron (B.1.1.529): SARS-COV-2 variant of concern. 2021; `https://www.who.int/news/item/26-11-2021-classification-of-omicron-(b.1.1.529)-sars-cov-2-variant-of-concern`.

(38) Geyer, C. J. Practical markov chain monte carlo. *Statistical science* **1992**, 473–483.

(39) Jaynes, E. T. Information theory and statistical mechanics. *Physical review* **1957**, *106*, 620.

(40) Jaynes, E. T. *Probability theory: The logic of science*; Cambridge university press, 2003.

(41) Pitera, J. W.; Chodera, J. D. On the use of experimental observations to bias simulated ensembles. *Journal of chemical theory and computation* **2012**, *8*, 3445–3451.

(42) Roux, B.; Weare, J. On the statistical equivalence of restrained-ensemble simulations with the maximum entropy method. *The Journal of chemical physics* **2013**, *138*.

(43) Amirkulova, D. B.; White, A. D. Recent advances in maximum entropy biasing techniques for molecular dynamics. *Molecular Simulation* **2019**, *45*, 1285–1294.

(44) Hadfield, J.; Megill, C.; Bell, S. M.; Huddleston, J.; Potter, B.; Callender, C.; Sagulenko, P.; Bedford, T.; Neher, R. A. Nextstrain: real-time tracking of pathogen evolution. *Bioinformatics* **2018**, *34*, 4121–4123.

(45) Ellis, D.; Brunette, N.; Crawford, K. H.; Walls, A. C.; Pham, M. N.; Chen, C.; Herpoldt, K.-L.; Fiala, B.; Murphy, M.; Pettie, D.; others Stabilization of the SARS-CoV-2 spike receptor-binding domain using deep mutational scanning and structure-based design. *Frontiers in immunology* **2021**, *12*, 710263.

(46) Gangavarapu, K.; Latif, A. A.; Mullen, J. L.; Alkuzweny, M.; Hufbauer, E.; Tsueng, G.; Haag, E.; Zeller, M.; Aceves, C. M.; Zaiets, K.; others Outbreak. info genomic reports: scalable and dynamic surveillance of SARS-CoV-2 variants and mutations. *Nature Methods* **2023**, *20*, 512–522.

(47) Khare, S.; Gurry, C.; Freitas, L.; Schultz, M. B.; Bach, G.; Diallo, A.; Akite, N.; Ho, J.; Lee, R. T.; Yeo, W.; others GISAID's role in pandemic response. *China CDC weekly* **2021**, *3*, 1049.

(48) Hoffmann, M.; Kleine-Weber, H.; Schroeder, S.; Krüger, N.; Herrler, T.; Erichsen, S.; Schiergens, T. S.; Herrler, G.; Wu, N.-H.; Nitsche, A.; others SARS-CoV-2 cell entry depends on ACE2 and TMPRSS2 and is blocked by a clinically proven protease inhibitor. *cell* **2020**, *181*, 271–280.

(49) Beyerstedt, S.; Casaro, E. B.; Rangel, É. B. COVID-19: angiotensin-converting enzyme 2 (ACE2) expression and tissue susceptibility to SARS-CoV-2 infection. *European journal of clinical microbiology & infectious diseases* **2021**, *40*, 905–919.

(50) Jackson, C. B.; Farzan, M.; Chen, B.; Choe, H. Mechanisms of SARS-CoV-2 entry into cells. *Nature reviews Molecular cell biology* **2022**, *23*, 3–20.

(51) Høie, M. H.; Cagiada, M.; Frederiksen, A. H. B.; Stein, A.; Lindorff-Larsen, K. Predicting and interpreting large-scale mutagenesis data using analyses of protein stability and conservation. *Cell reports* **2022**, *38*.

(52) Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Killeen, T.; Lin, Z.; Gimelshein, N.; Antiga, L.; others Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems* **2019**, *32*.

(53) Agarap, A. F. Deep learning using rectified linear units (relu). *arXiv preprint arXiv:1803.08375* **2018**,

(54) Kingma, D. P. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* **2014**,

(55) Pless, D. D.; Lennarz, W. J. Enzymatic conversion of proteins to glycoproteins. *Proceedings of the National Academy of Sciences* **1977**, *74*, 134–138.

(56) Alley, E. C.; Khimulya, G.; Biswas, S.; AlQuraishi, M.; Church, G. M. Unified rational protein engineering with sequence-based deep representation learning. *Nature methods* **2019**, *16*, 1315–1322.

(57) Rao, R.; Bhattacharya, N.; Thomas, N.; Duan, Y.; Chen, P.; Canny, J.; Abbeel, P.; Song, Y. Evaluating protein transfer learning with TAPE. *Advances in neural information processing systems* **2019**, *32*.

(58) Heinzinger, M.; Elnaggar, A.; Wang, Y.; Dallago, C.; Nechaev, D.; Matthes, F.; Rost, B. Modeling aspects of the language of life through transfer-learning protein sequences. *BMC bioinformatics* **2019**, *20*, 1–17.

(59) Elnaggar, A.; Heinzinger, M.; Dallago, C.; Rehawi, G.; Wang, Y.; Jones, L.; Gibbs, T.; Feher, T.; Angerer, C.; Steinegger, M.; others Prottrans: Toward understanding the language of life through self-supervised learning. *IEEE transactions on pattern analysis and machine intelligence* **2021**, *44*, 7112–7127.

(60) Rives, A.; Meier, J.; Sercu, T.; Goyal, S.; Lin, Z.; Liu, J.; Guo, D.; Ott, M.; Zitnick, C. L.; Ma, J.; others Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences. *Proceedings of the National Academy of Sciences* **2021**, *118*, e2016239118.

(61) Barros, E. P.; Casalino, L.; Gaieb, Z.; Dommer, A. C.; Wang, Y.; Fallon, L.; Raguette, L.; Belfon, K.; Simmerling, C.; Amaro, R. E. The flexibility of ACE2 in the context of SARS-CoV-2 infection. *Biophysical journal* **2021**, *120*, 1072–1084.

(62) Roe, D. R.; Cheatham III, T. E. PTRAJ and CPPTRAJ: software for processing and analysis of molecular dynamics trajectory data. *Journal of chemical theory and computation* **2013**, *9*, 3084–3095.

(63) McGibbon, R. T.; Beauchamp, K. A.; Harrigan, M. P.; Klein, C.; Swails, J. M.; Hernández, C. X.; Schwantes, C. R.; Wang, L.-P.; Lane, T. J.; Pande, V. S. MDTraj: a modern open library for the analysis of molecular dynamics trajectories. *Biophysical journal* **2015**, *109*, 1528–1532.

(64) Ke, G.; Meng, Q.; Finley, T.; Wang, T.; Chen, W.; Ma, W.; Ye, Q.; Liu, T.-Y. Lightgbm: A highly efficient gradient boosting decision tree. *Advances in neural information processing systems* **2017**, *30*.

TOC Graphic

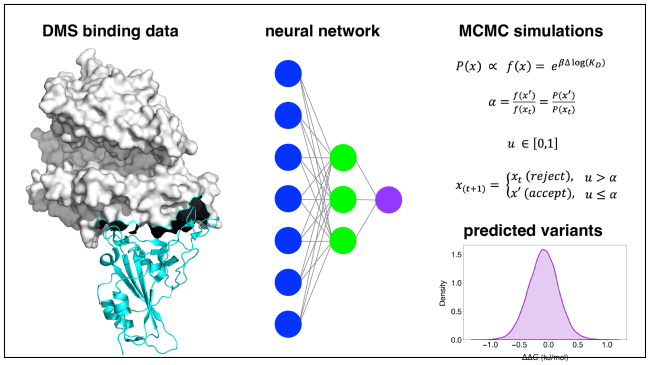
DMS binding data
neural network
MCMC simulations
$P(x) \propto f(x) = e^{\beta \Delta \log(K_D)}$
$\alpha = \frac{f(x')}{f(x_t)} = \frac{P(x')}{P(x_t)}$
$u \in [0,1]$
$x_{(t+1)} = \begin{cases} x_t\ (reject), & u > \alpha \\ x'\ (accept), & u \leq \alpha \end{cases}$
predicted variants
Density
1.5
1.0
0.5
0.0
−1.0
−0.5
0.0
0.5
1.0
ΔΔG (kcal/mol)

Table 1: Summary of Variants of concern. Mutations predicted by the simulation are shown in bold. Omicron variants names are also labeled with Nextstrain clade scheme.[44]

Name	Strain	RBD mutations
Alpha	B.1.1.7	**E484K, N501Y**
Beta	B.1.351	K417N, **E484K, N501Y**
Gamma	P1	K417T, **E484K, N501Y**
Delta	B.1.617.2	**L452**R, T478K
Epsilon	B.1.42(7/9)	**L452**R
Eta	B.1.525	**E484K**
Iota	B.1.526	**E484K**
Iota'	B.1.526	**S477N**
Kappa	B.1.617.1	**L452**R, **E484**Q
Zeta	P2	**E484K**
Mu	B.1.621(.1)	R346K, **E484K**, **N501Y**
21K Omicron	BA.1	**G339D**, S371L, S373P, S375F, K417N, **N440K**, G446S, **S477N**, T478K, **E484**A, **Q493**R, G496S, **Q498**R, **N501Y**, **Y505**H
21L Omicron	BA.2	**G339D**, S371F, S373P, S375F, T376A, D405N, R408S, K417N, **N440K**, **S477N**, T478K, **E484**A, **Q493**R, **Q498**R, **N501Y**, **Y505**H
22A/B Omicron	BA.4/5	**G339D**, S371F, S373P, S375F, T376A, D405N, R408S, K417N, **N440K**, **L452**R, **S477N**, T478K, **E484**A, F486V, **Q498**R, **N501Y**, **Y505**H
22D Omicron	BA.2.75	**G339**H, S371F, S373P, S375F, T376A, D405N, R408S, K417N, **N440K**, G446S, **N460K**, **S477N**, T478K, **E484**A, **Q498**R, **N501Y**, **Y505**H
22E Omicron	BQ.1	**G339**D, S371F, S373P, S375F, T376A, D405N, R408S, K417N, **N440K**, K444T, **L452**R, **N460K**, **S477N**, T478K, **E484**A, F486V, **Q498**R, **N501Y**, **Y505**H
22F Omicron	XBB	**G339**H, R346T, L368I, S371F, S373P, S375F, T376A, D405N, R408S, K417N, **N440K**, V445P, G446S, **N460K**, **S477N**, T478K, **E484**A, F486S, F490S, R493Q, **Q498**R, **N501Y**, **Y505**H
23C Omicron	CH.1.1	**G339**H, R346T, S371F, S373P, S375F, T376A, D405N, R408S, K417N, **N440K**, K444T, G446S, **L452**R, **N460K**, **S477N**, T478K, **E484**A, R493Q, **Q498**R, **N501Y**, **Y505**H