Quantified Dynamics-Property Relationships: Data-Efficient Protein Engineering with Machine Learning of Protein Dynamics

Tucker Emme Burgin*

Thayer School of Engineering, Dartmouth College, Hanover, NH

E-mail: emme.burgin@dartmouth.edu

Abstract

Machine learning has proven to be very powerful for predicting mutation effects in proteins, but typically only when a large amount of training data is available. Because experiments to collect training data are often expensive, time-consuming, and/or otherwise limited, alternatives that make good use of small amounts of data to guide protein engineering are of high potential value. One potential alternative to large-scale benchtop experiments for collecting training data is high-throughput molecular dynamics simulation; however, to date this source of data has been largely absent from the literature. Here, I introduce a new method for selecting desirable protein variants based on quantified relationships between a small number of experimentally determined labels and descriptors of their dynamic properties. These descriptors are provided by deep neural networks trained on data from molecular dynamics simulations of variants of the protein of interest. I demonstrate that this approach can obtain very highly optimized variants based on small amounts of experimental data, greatly outperforming machine learning-guided directed evolution with the same amount of experimental data. Furthermore, I show that quantified dynamics-property relationships based on

only a handful of experimentally labeled example sequences can be used to accurately predict the key residues that are most relevant to determining the property in question, even when that information could not have been known or predicted based on either the molecular dynamics simulations or the experimental data alone. This work establishes a new, general, and highly practical tool for incorporating protein dynamics information to guide data-efficient protein engineering.

Introduction and Background

Protein engineering has been greatly accelerated by the application of machine learning (ML) to build mutation effect prediction models: tools trained to accurately predict the downstream effects of one or more mutations on protein functions of interest, such as substrate specificity, thermostability, binding affinity, or any other parameter.[1] Despite the promise of this emerging field, a principal limitation of ML for predicting mutation effects is the availability of appropriate training data. Even highly specialized models making predictions of the effects of low numbers of mutations per protein variant typically require at least hundreds or thousands of high-quality training examples,[2] which can be prohibitively expensive to obtain. This issue is further compounded for larger numbers of mutations per variant, and models may need to be partially or entirely retrained on new data once more than a handful of mutations have been applied to the initial protein sequence.

Large-scale experimental sequencing approaches like deep mutational scanning have made progress towards meeting the need for large amounts of labeled protein sequence data, but these remain expensive, limited by available transfection and sequencing technologies, and restricted to targets where an appropriate functional assay is available.[3] Accordingly, a fairly large body of literature has been published in recent years aimed at developing more efficient ML tools for obtaining better variants through fewer experiments. Many strategies have been proposed, but one major commonality is the application of additional information to guide the selection of training data, such as through unsupervised clustering of candidate

sequences to diversify training data,[4] pre-filtering based on predictions of structural stability,[5,6] or through the use of protein language models or encodings trained on large numbers of unlabeled protein sequences.[7–10] Despite the improved performance of all of these approaches over unguided directed evolution or simple ML models, in practice the number of experiments required to obtain highly optimized variants remains high in most cases, with hundreds of experiments still needed to engineer even small proteins with highly constrained sets of permitted mutations.

In addition to the difficulty of collecting appropriate training data, existing mutation effect prediction ML models lack the ability to offer mechanistic insight, and operate as a black box. In other words, no matter how accurate a model may be at predicting the effect of a mutation on protein function, its impact is limited by the absence of explanations as to *why* those effects arise. If mutation effect prediction models were capable of reporting molecular-level details about the relationship between sequence and function, this would have a potentially transformative impact on the ability of researchers to understand and engineer protein behavior.

Here, I develop an approach to protein mutation effect prediction that can obtain highly optimized protein variants within only a handful of experimental measurements (tens instead of hundreds), while also providing molecular-level explanations of those effects. This approach is based on combining traditional mutation effect prediction models with dynamic, biophysical information obtained from high-throughput molecular dynamics simulations of protein variants. This method is an evolution of quantified structure-property relationship (QSPR) modeling that I call quantified dynamics-property relationship (QDPR) modeling. I first explain the rationale and design of the method, and then demonstrate that this approach can produce high-fitness variants with very small experimental budgets across two highly distinct proteins and functions: the *Streptococcus* protein G B1 domain (GB1) and its affinity for binding human IgG, and *Aequorea victoria* green fluorescent protein (*Av*GFP) fluorescence intensity. Then, I show how predictions of protein dynamics provide a frame-

work for interpreting the experimental data in the context of physically meaningful molecular quantities. Excitingly, the model is able to identify key residues mediating protein-protein binding interactions based on simulations of variants of just one protein, by correlating predictions of molecular-level dynamic effects with benchtop binding affinity measurements for just 16 sequences. This work establishes a new and powerful alternative to the collection of large experimental datasets for training protein mutation effect models, opens a new avenue for studying the dynamic basis of protein function based on small amounts of mutation effect data, and addresses a major gap in the field by augmenting ML-guided protein engineering with high-throughput atomistic molecular dynamics simulations that do not need to directly measure the engineered property.

Methods

Technical rationale

The general outline for the methodology used in this work is shown in Fig. 1. Typical approaches to guiding protein engineering using neural networks focus on models trained directly to predict the functional label of interest (Fig. 1a). This type of model is incorporated here as well, but the central methodological innovation in this work lies in the extraction of large numbers of biophysical features from unbiased molecular dynamics simulations of randomly selected protein variants (Fig. 1b); the training of neural networks to predict each of these biophysical features based on protein sequences (Fig. 1c); and the evaluation of correlations between experimentally determined measurements of protein variant properties and the predictions of each network (Fig. 1f), which are used to guide the selection of enhanced variants (Fig. 1e).

Importantly, the simulations themselves are *not* constructed so as to attempt to produce any direct measurement or prediction of the protein property to be engineered. For example, when engineering a protein for improved binding affinity for a particular partner, the binding

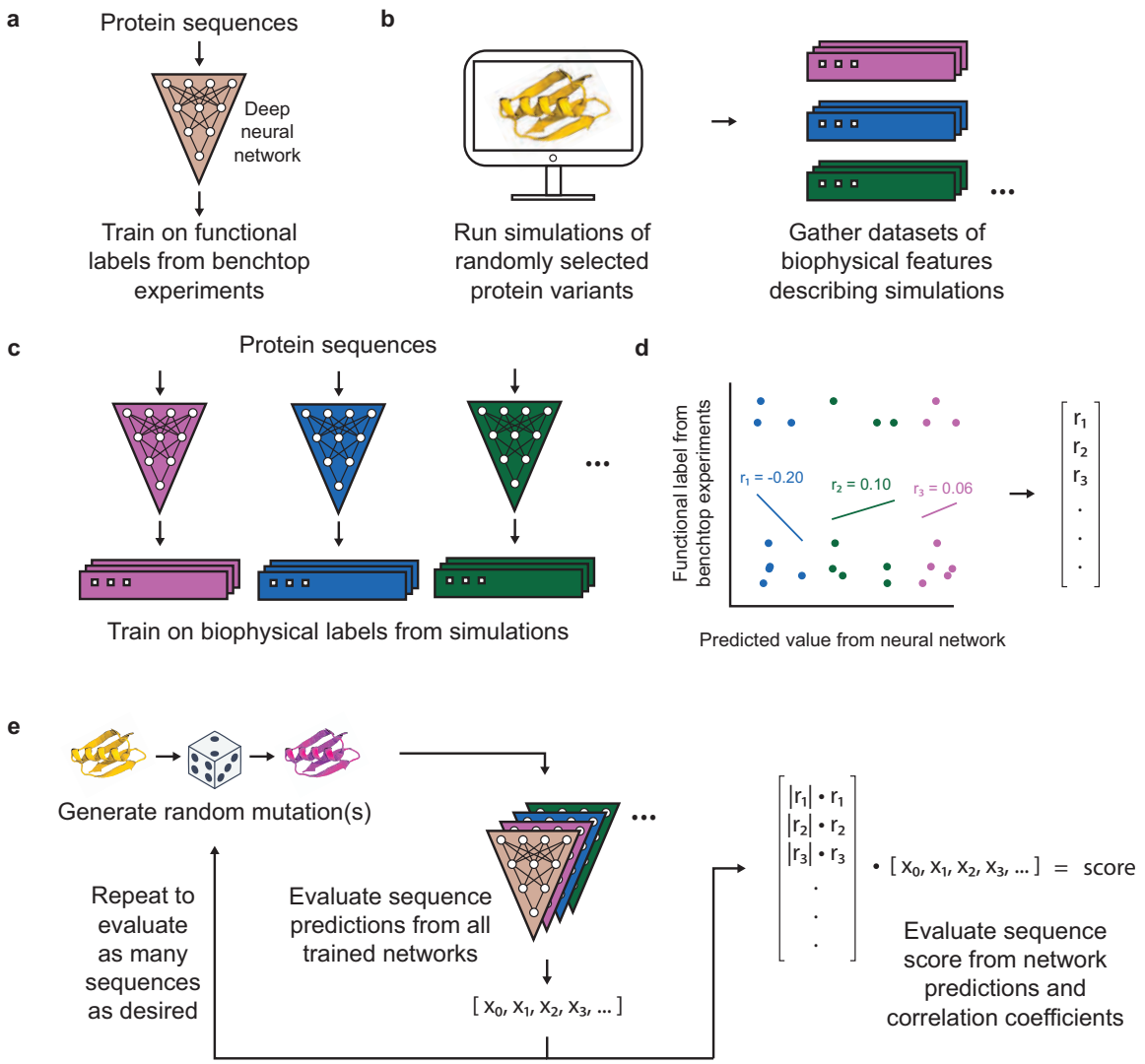

Figure 1: **Overview of the method described in this work.** **(a)** Training a deep neural network on functional labels describing the property of some protein to be engineered. This represents a typical approach to guiding protein engineering with machine learning. The three-layer neural network shown in the figure is representative only; in this work, convolutional neural networks were used (see Methods > Training of neural networks). **(b)** Extracting datasets of biophysical features that describe the unbiased simulations of hundreds-to-thousands of randomly selected protein variants. Each color represents a different type of biophysical feature. **(c)** Using those biophysical feature datasets to train a set of additional deep neural networks to make predictions of each feature based on protein sequences. **(d)** Evaluating the correlation between each predicted biophysical feature and a small set of known, experimentally determined labels. The correlation coefficients are saved in a vector and are taken as a measure of the relative importance of each feature. **(e)** Generating random mutations and then evaluating them using each of the trained neural networks. The predictions from each network are saved as a vector, whose dot product with the square correlation vector from the previous step is interpreted as a score that can be compared across different protein variants to guide selection of desirable mutations. Evaluating the network predictions is fast, so large numbers of possible variants can be efficiently screened.

partner was not included in the simulations at all. Instead, the purpose of the simulations is to collect data describing the molecular biophysical properties of the protein itself, which are then correlated with experimental measurements of the target property. The hypothesis guiding this design is that improved protein variants can be selected by biasing the selection process towards variants whose biophysical properties are correlated with desired changes in the target property. This approach also minimizes the amount of information that needs to be known beforehand about the molecular basis of the desired protein property; e.g., it is not necessary to know where or how a partner binds to the target protein in order to apply this method to optimize binding affinity.

Molecular dynamics simulations

Molecular dynamics simulations were performed using Amber 20.[11] The GB1 model was based on PDB ID: 1PGA[12] and the *Av*GFP model was based on PDB ID: 2WUR.[13] Proteins were modeled in the wild type using the ff19SB force field[14] and an octagonal box of OPC3 water molecules[15] with a minimum distance between the protein and the edge of the box of 8 Å. The H++ webserver[16] was used to determine initial protonation states of titratable residues. Models were minimized over 5000 steps and then heated at constant volume from 100 to 303.15 K over 11,000 2-fs steps. Hydrogen mass repartitioning[17] with a factor of 3 was applied to enable a longer simulation time step, and the models were equilibrated at constant pressure over 250,000 4-fs time steps. Based on these initial equilibrated models, mutated models were generated using PyRosetta[18] to apply mutations and PROPKA3[19] to assess changes in protonation states before being re-minimized and heated using the same workflow as for the wild type model. Mutations were randomly selected, with an equal probability across the whole protein of either one or two mutations for GB1 or anywhere between one and seven mutations for *Av*GFP (the residues making up the chromophore in GFP were not subject to mutation). Production simulations for each variant were run for 100 ns (arbitrarily, in a single, continuous trajectory for GB1 or as two, independent 50-ns

trajectories for *Av*GFP).

Extraction of biophysical features from simulation

Simulations were analyzed using pytraj,[20] MDTraj,[21] and MDAnalysis.[22,23] Before analysis, the first 10% of each trajectory was discarded to allow for equilibration. Three different types of biophysical features were extracted from each GB1 simulation: by-residue root-mean-square fluctuation (RMSF), by-residue Kabsch-Sander backbone hydrogen bonding energy,[24] and the relative weights of the projections of each mutant trajectory onto each of the first 70 components of a PCA decomposition of the motion of alpha carbons from an arbitrarily selected representative trajectory for that protein. Additionally, for the *Av*GFP simulations, by-residue Wernet-Nilsson hydrogen bonding energies[25] and by-residue global allosteric communication scores[26] were included as features. These features served as training data for the neural networks in the next step.

Training of neural networks

Convolutional neural networks for each protein were trained to predict each of the biophysical features based on the methodology, network architectures, and hyperparameters described by Gelman et al.[2] As in that work, protein sequences were encoded using a combined one-hot and biochemical properties encoding based on the amino acid index database, AAindex1.[27] Each biophysical feature network was trained on data from the same set of 500 (for GB1) or 1500 (for *Av*GFP) MD simulations, with 1000 or 21 samples, respectively, reserved for validation. Models were trained until the validation mean squared error stopped decreasing, with a patience of 500 epochs, and restoring the model weights with the lowest validation error at the end. This process was repeated three times per model, with only the lowest-error model against the validation set saved for later use. Label values were linearly rescaled to between 0 and 1 before training. Performance benchmarks for each of the trained models are available as Supplementary file pearson_scores.xlsx.

Neural networks trained to directly predict the experimental labels were trained in the same manner, except with a variable validation set size reflecting 10% of the total available data and a patience equal to the total number of steps for which the corresponding model was trained in Gelman et al.[2] Overall better performance was observed with this approach compared to using no validation set with a fixed number of training epochs.

Machine learning-guided protein engineering

Taking advantage of very large published datasets for mutation effects in GB1[28] and *Av*GFP,[29] protein engineering campaigns (successive rounds of variant selection) were simulated by drawing labeled sequences from databases in place of conducting experiments directly. Selections of an arbitrary fixed number of 8 (for GB1) or 16 (for *Av*GFP) sequences were drawn for each round, with a different random selection used for the zeroth round of each campaign. For each protein, three strategies were compared: fully random selection; an ML-guided directed evolution approach based on the predictions from a model trained on the experimental labels of sequences that had been selected in the previous rounds; and the new QDPR ML-guided approach that incorporated the biophysical feature predictions from the neural networks trained on simulation data. For each case, 100 independent campaigns were performed in order to characterize a range of possible outcomes.

The QDPR approach starting from the trained models is schematized in Fig. 1d,e. First, each of the sequences from the set of experiments that had been selected up to that point were passed through each of the biophysical prediction networks and the functional label network. The correlation of a linear regression between the resulting feature predictions and the experimental labels was measured, with the square correlation coefficients of each feature saved to a vector $\mathbf{R}$ (values in the range [-1, 1], computed as $|r| \cdot r$ (to preserve sign) where r is the Pearson's correlation coefficient). Then, a random, as-of-yet unseen variant from the experimental dataset was selected (without exposing its corresponding label to the model) and also passed through each of the neural networks. The resulting predictions were saved

in a prediction vector $\mathbf{X}$, with positions corresponding to those of the feature correlation coefficients in the previous vector. The values of the prediction vector were doubled and 1 was subtracted to bring them into the same range of [-1, 1] (so as to weight very small and very large predictions equally). Finally, a score was computed for the sequence according to the dot product $\mathbf{R} \cdot \mathbf{X}$. In each round, all of the sequences for which experimental labels were available were scored in this way (536,084 for GB1 and 45,205 for *Av*GFP, less however many had already been selected up to that point), and the top 8 or 16 scoring sequences were then selected, respectively. Then, their real experimental labels were taken from the dataset (simulating the process of conducting experiments on the selected sequences) to serve as further training data for the next round. The set of experimental labels of the selected sequences from each round was saved for later analysis.

Results and Discussion

Simulated protein engineering campaigns

The aggregated results of the simulated protein engineering campaigns are shown in Fig. 2. QDPR was the standout best performing method compared to random search or ML-guided directed evolution in both proteins. In addition to being very different proteins and target properties, the overall distributions of fitness values are very different: the GB1 dataset is unimodal with a very long tail of high-performing variants, whereas the *Av*GFP dataset is bimodal with a large number of poor-performers and a relatively short tail of higher performers. It is unsurprising that ML-guided directed evolution performed poorly, because much larger amounts of experimental data would be necessary to train strong models; however, the inclusion of models trained on simulation data successfully overcame this limitation for both proteins. In both proteins, only QDPR eventually raised the interquartile range of sampled sequence scores above the wild type score.

Because in practice the desired outcome of a protein engineering campaign is usually a

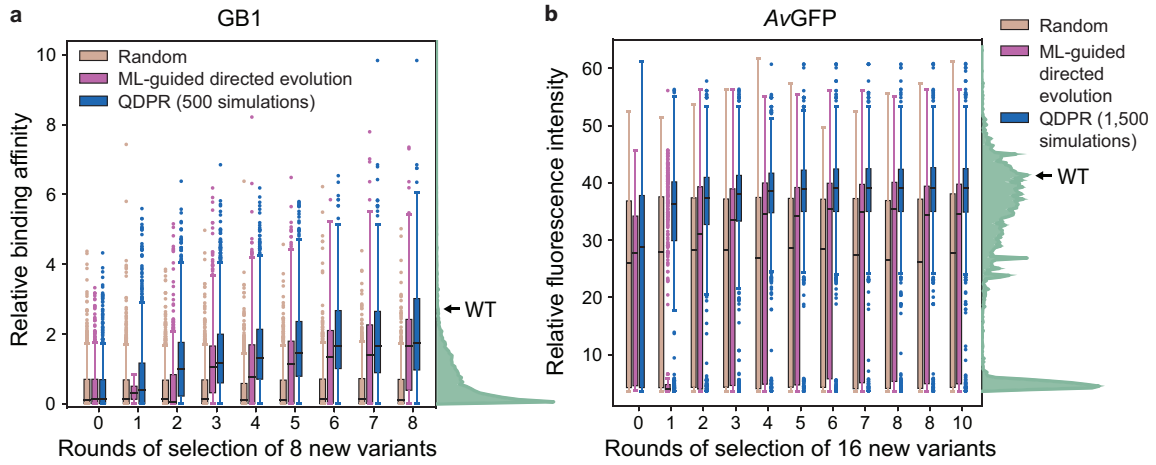

Figure 2: **Aggregated results of simulated protein engineering campaigns.** Box-and-whisker plots represent all of the sets of 8 (for GB1, **(a)**) or 16 (for *Av*GFP, **(b)**) sequences selected in each round (not the cumulative set of all sequences theretofore selected), aggregated across all 100 campaigns for each strategy (i.e., 800 or 1600 sequences per round, respectively). Each box extends from the first to the third quartile of the data, with a line at the median and whiskers extending 1.5 times the interquartile range from the box. Individual points are shown where samples lie outside this range. Random search is tan; ML-guided directed evolution is magenta; QDPR is blue. Histograms at the right of each plot represent the full distributions of available experimental labels for each protein, with wild type values indicated.

single optimized variant, I also looked at the likelihood of obtaining variants in the 99.9th percentile of the datasets with each method, as well as distributions of the highest-fitness variants achieved in each of the 100 campaigns. These are shown in Fig. 3 and Fig. 4, respectively. In the case of *Av*GFP, the median highest fitness scores obtained for random, ML-guided directed evolution, and QDPR campaigns was 50.7, 52.6, and 55.9, respectively. For GB1, the medians were 2.99, 3.36, and 4.28, respectively.

For comparison to another method already published that takes advantage of similar amounts of training data, Biswas *et al.* demonstrated a machine learning-based low-N approach for *Av*GFP engineering using the UniRep protein language model refined on evolutionarily related sequences, called eUniRep.[8] In that work, 24 or 96 experimental examples were used to tune the selection of variants, with the figure of merit given as a "hit rate" – the fraction of variants suggested by the model with fluorescence intensity above that of wild type *Av*GFP. The hit rates reported for two independent campaigns were 10.9% and 5.5%

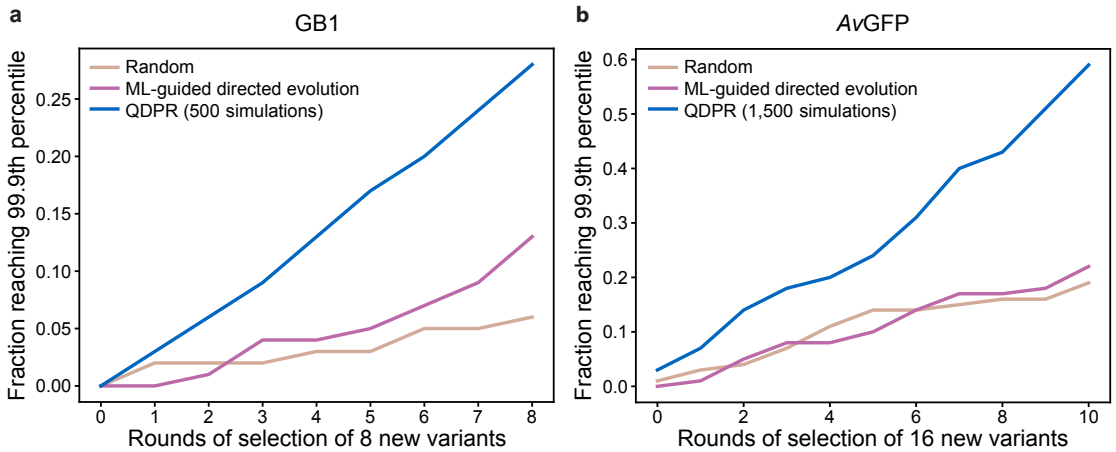

Figure 3: **Cumulative likelihood of 99.9th percentile variants.** Each line represents the observed likelihood of achieving a variant in the 99.9th fitness percentile by a given round using a given method for **(a)** GB1 and **(b)** *Av*GFP. Random search is tan; ML-guided directed evolution is magenta; QDPR is blue. The cutoff fitness values for the 99.9th percentile were approximately 5.05 for GB1 and 55.04 for *Av*GFP.

for 24 training examples and 12.5% and 10.6% for 96 training examples, respectively. By comparison, the hit rate (defined in the same way) for *Av*GFP QDPR campaigns averaged 14.3% based on 16 training examples (that is, in the first round) and 23.8% in the round with access to 96 training examples. The full hit rate data is provided in Supplementary Fig. 1.

To assess the ability of these results to scale beyond 10 steps of variant selection, the campaigns of GB1 engineering were repeated another 25 times for 40 rounds of selection each. The QDPR approach continued to consistently outperform the ML-guided directed evolution approach, both in terms of frequency of resulting in extremely high-fitness variants, and in terms of median highest fitness achieved over the course of the campaigns. These data are available in Supplementary Fig. 2.

Quantified feature importance

Besides improved variant selection, another key advantage of QDPR is that the observed correlations between each biophysical feature and experimentally determined target labels

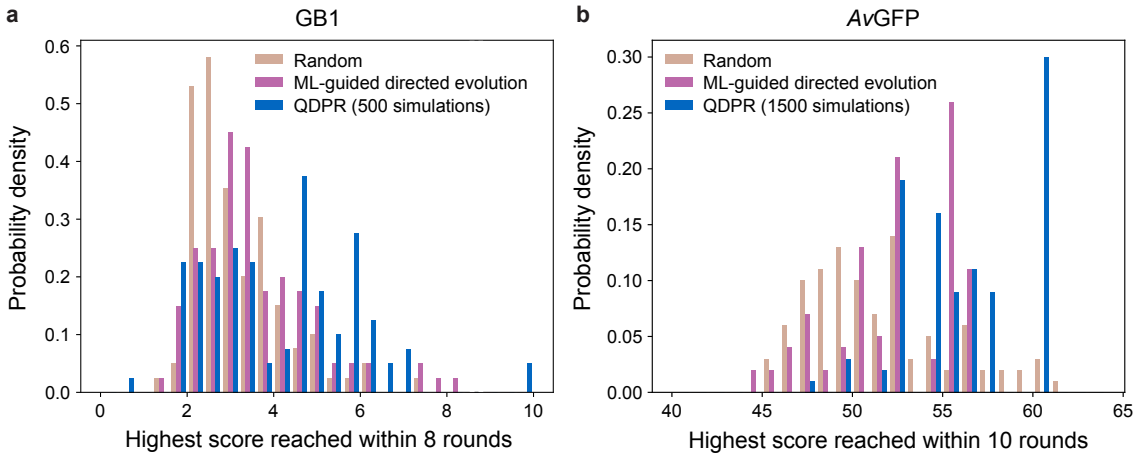

Figure 4: **Histograms of highest fitness values achieved across 100 independent campaigns.** Histograms are provided for **(a)** GB1 and **(b)** *Av*GFP. Random search is tan; ML-guided directed evolution is magenta; QDPR is blue.

can be used to infer the relative importance of each biophysical feature in the molecular basis of the target protein function. For example, in the case of GB1 binding to human IgG, the bound structure is known (see PDB ID: 1FCC[30]), but IgG was not present in the simulations, providing a test case where the predicted importance of each biophysical feature can be evaluated. To illustrate this, a consensus ranking of by-residue RMSF and backbone hydrogen bonding energy features was computed after only the first round of variant selection based on pairwise preferences across each of the 100 QDPR campaigns of GB1 engineering. Based on this, a relative importance score was assigned to each residue representing the ranks of features describing that residue. This consensus ranking approach is designed to obtain the "most typical" ranking to be expected from any given campaign (as opposed to an average taking advantage of data all 100 campaigns), and is visualized in Fig. 5. More details of this analysis are reported in the Supplementary Information.

Excitingly, many residues with very high importance ranks turn out to be the ones most directly responsible for IgG binding, meaning that even if the GB1 residues that associate with IgG had not been known in advance, they could have been accurately predicted using this approach. Other high-importance residues are hydrophobic residues with their side chains packed inside the protein, with clear roles to play in fold stability. Besides crystal-

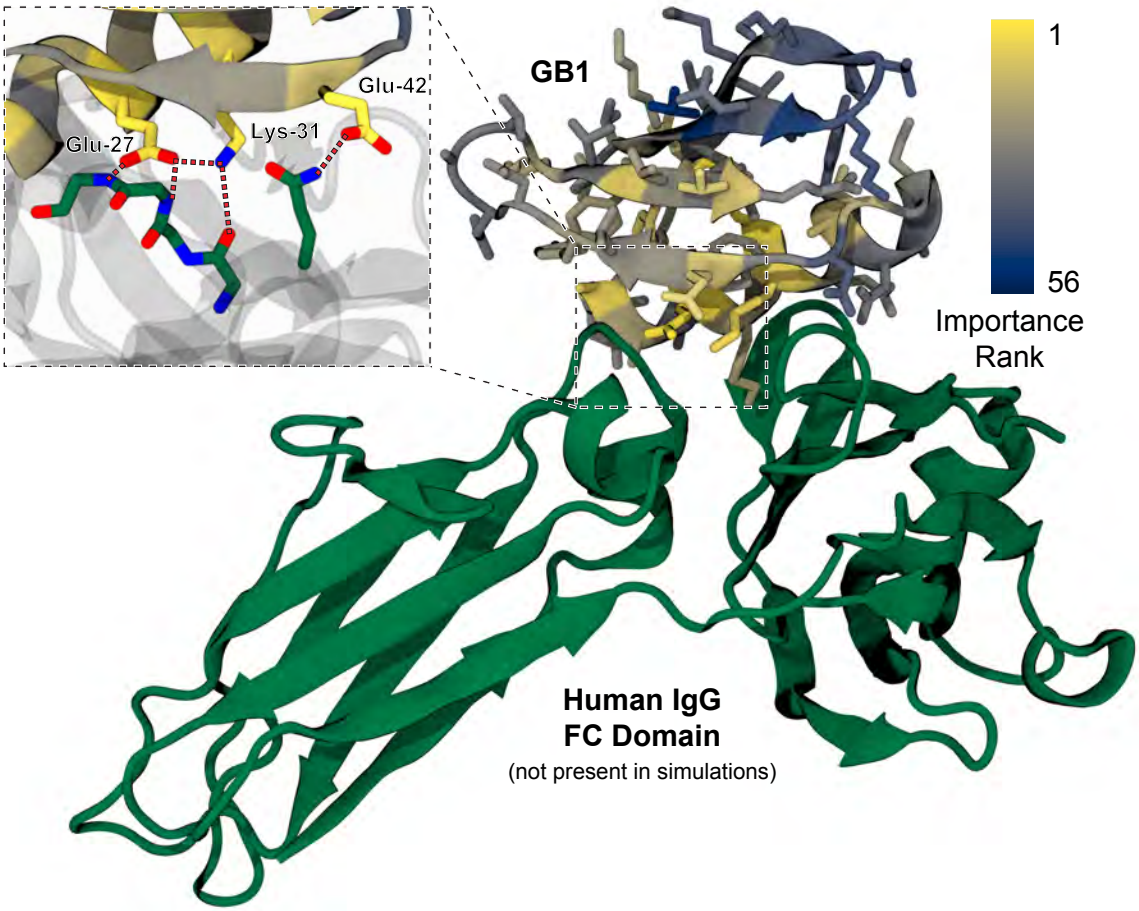

Figure 5: **Visualization of residue importance ranks.** A portion of crystal structure PDB ID: 1FCC[30] depicting binding between GB1 the fragment crystallizable (FC) domain of human IgG in green. GB1 is colored by-residue based on the relative importance rank of each position's RMSF and backbone hydrogen bonding energy features in the consensus ranking of features among the 100 QDPR 500-simulation campaigns after a single round of variant selection each. IgG was not included in the simulations, which were of variants of GB1 in water only. Inset, three of the highest-ranked residues – Glu-27, Lys-31, and Glu-42 – are shown to form a key hydrogen bonding network that mediates binding between the two proteins.

lographic evidence, the importance of the identified residues for binding is reflected in the large effect associated with mutating them: whereas the average relative log binding affinity score across all variants in the GB1 dataset is -2.42 ± 2.28 (n = 536,084), the average for all variants wherein either Glu-27 or Lys-31 is mutated is -5.47 ± 0.914 (n = 38,625). Furthermore, the importance of these residues was frequently correctly identified by correlation of biophysical features with just 16 experimentally labeled sequences, even in campaigns where none of those 16 sequences included mutations in either position (which was the vast majority). Instead, changes in the RMSF or backbone hydrogen bonding energy of these residues were observed in most cases only as consequences of mutations elsewhere in the protein during MD simulations, highlighting the importance of features trained on MD simulations in making accurate predictions with small amounts of experimental data. A similar visualization for *Av*GFP is available as Supplementary Fig. 3.

Mapping explored regions of fitness landscapes

One of the principal advantages of ML-guided protein engineering in general is the ability to jump over local fitness minima separating more desirable regions of fitness landscapes, where methods like directed evolution often get stuck.[31] Accordingly, any ML-informed approach ought to explore a broad search space, rather than seeking to obtain optimized variants only through exploitation of a local region. Although the incorporation of biophysical feature information obtained from molecular dynamics simulations already provides compelling theoretical justification for the improved performance observed using QDPR compared to an ML approach based only on scant experimental data, it is not *a priori* clear what effect this might have on the regions of the fitness landscape that QDPR is likely to sample from.

To study this, I computed a pairwise Hamming distance matrix of all of the sequences in the full *Av*GFP dataset of scored variants and then used multi-dimensional scaling to project the fitness landscapes onto two dimensions for visualization. A comparison of the overall fitness landscape with the sampling coverage of ML-guided directed evolution and

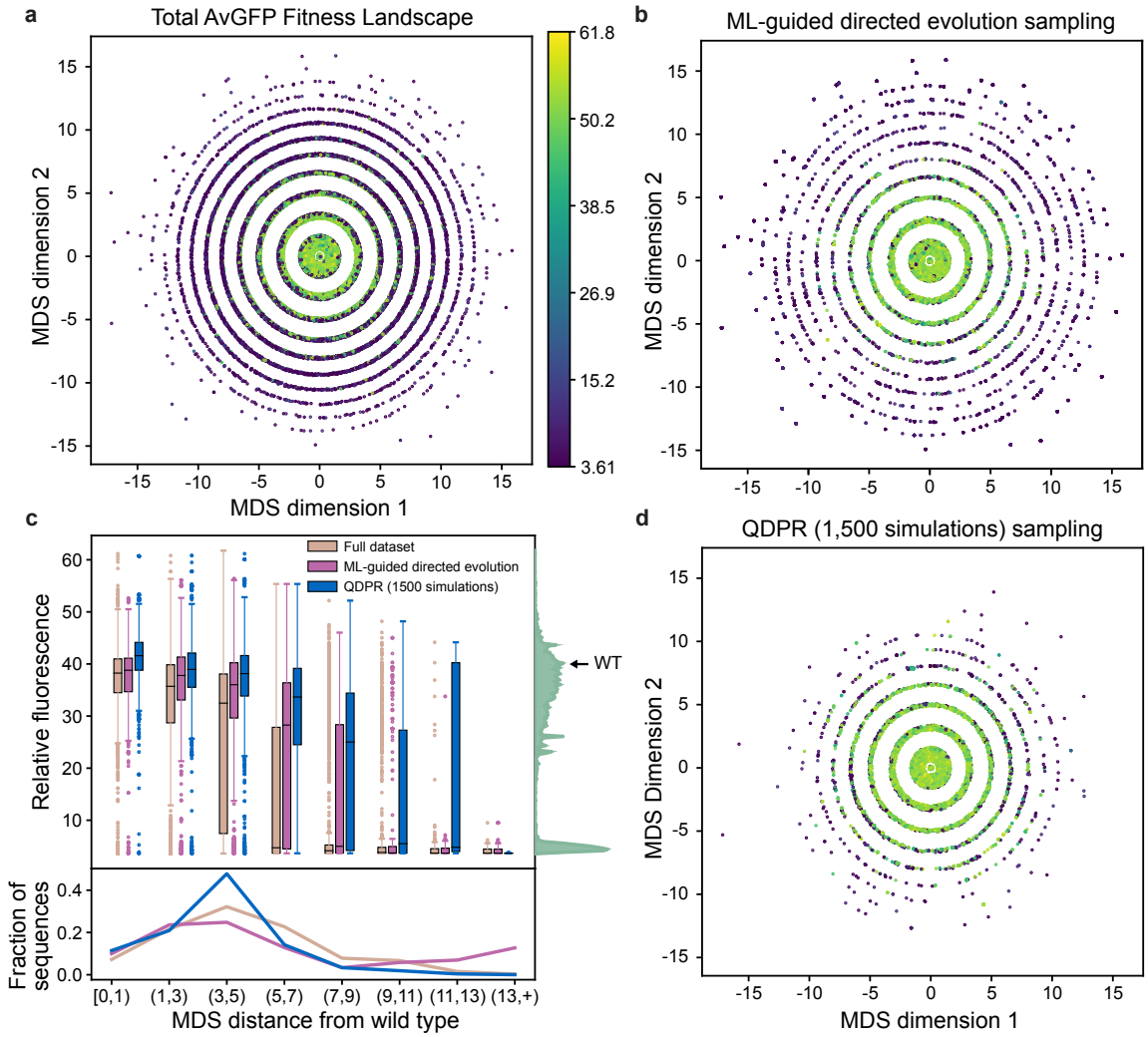

Figure 6: **2D fitness landscape and sampling coverage for *Av*GFP. (a)** Fitness landscape of the overall dataset. Each point represents a unique sequence, colored according to experimentally determined fitness score. The concentric structure of the landscape is a consequence of the wide range of numbers of mutations in each variant relative to the wild type (at 0, 0). **(b)** Breakdown of the full dataset and the aggregate sampling from the 100 ML-guided directed evolution and QDPR campaigns, respectively, binned by distance along the first two MDS dimensions from the wild type sequence. The box plots represent the distributions of fitness scores in each bin, with a line at the median, the box representing the interquartile range, and whiskers extending to the furthest datapoint within 1.5 times the interquartile range beyond. The histogram at right is the overall distribution of fluorescence values in the dataset. The lines at bottom represent the fraction of the population in each set that was sorted into each bin. **(c)** and **(d)** The aggregate samples from the 100 ML-guided directed evolution and QDPR campaigns, respectively, on the same axes and color bar as the overall landscape. The same total number of points appear in each of these two panels.

QDPR is shown in Fig. 6. Sampling remains highly distributed across sequence space when using QDPR, but is enriched in the regions of the fitness landscape densest with high-fitness variants (especially within 3-to-5 units away from the wild type, which is the region containing the highest density of high-fitness variants) and correspondingly reduced in low-fitness regions. However, QDPR is still able to identify some relatively high-fitness variants from regions dominated by low-fitness variants (i.e., MDS distance from wild type > 7) that are absent from ML-guided directed evolution sampling, suggesting that QDPR improves discrimination between high- and low-fitness variants across the fitness landscape.

Opportunities for further improvement

Although the results presented here already demonstrate significant improvement over straightforward implementations of ML-guided protein engineering informed only by benchtop experimental results, there remain many unanswered questions and opportunities for further improvement of this method. Many of the parameters and decisions made in obtaining these results were somewhat arbitrary, or otherwise chosen for the sake of simplicity. For example, the length and number of the simulations may be adjustable either up or down in order to obtain more accurate biophysical feature prediction models or to reduce the upfront computational cost before model training. Additionally, there is no reason to expect that the short list of types of biophysical features included in this work represents a complete or optimal description of the relevant biophysical information present in the simulations, and more features could certainly be included to further improve the results and expand the range of features available for interpretation of the molecular basis of experimental observations. At an even more fundamental level, the choice to rank the importance of each feature using the correlation coefficient from a linear regression neglects more nuanced relationships that may exist between features and experimental observations; for example, cases where a feature has an optimal value and deviations in either direction are deleterious. Because of the very large number of parameters and algorithms that may be improved upon, a more thorough

exploration of these questions is left for the future, and significant further improvements in the performance of this approach are almost certainly possible.

One possible extension of this method that bears special mention is the incorporation of active learning, both in the selection of variants for simulations and in the guiding of selection of variants during the subsequent engineering campaign. Active learning is a subset of ML based on quantifying model uncertainty so as to guide the collection of further model training data that will have the maximum marginal positive impact on model performance. Recent work out of the lab of Frances Arnold has demonstrated that incorporating active learning into an ML-guided directed evolution campaign greatly reduces the number of rounds required to obtain optimized results, and also improves the final optimized sequences compared to ML-guided directed evolution without active learning.[32] Similarly, active learning could probably be applied directly to QDPR to improve performance without increasing the number of simulations or experiments required.

QDPR in the broader context of ML-guided protein engineering methods

Researchers interested in applying the latest ML methods to improve protein engineering efforts now have a very wide array of choices.[33,34] QDPR features three interrelated advantages compared to other methods that have been published to date:

1. **Data efficiency.** Most ML-guided protein engineering methods are focused on either improved performance with large amounts of training data, or on the other extreme, zero-shot predictions (that is, made with no relevant labeled training data). QDPR makes efficient use of small numbers of labeled sequences, which are frequently attainable in many cases where large experimental datasets are not available.

2. **Quantified feature importance scores.** QDPR is unique among available methods in the ability to provide quantified, molecular-level dynamics predictions to ex-

plain *why* any given sequence is highly (or poorly) scored. Additionally, any type of molecular descriptor that can be obtained from an MD simulation can be included in this quantification, without requiring any additional data collection. This provides a throughline between ML-guided and rational protein engineering approaches, which are often treated as mutually exclusive alternatives in practice.

3. **QDPR leverages dynamics.** Molecular-level dynamics information has been more-or-less totally excluded from state-of-the-art protein engineering approaches to date, despite ample evidence that mutation effects often cannot be explained based only on static structures.[33,35] QDPR goes far beyond broad-strokes descriptions of conformational ensembles[36–38] and instead leverages atomistic simulation data to accurately predict mutation effects on high-resolution dynamic features. Besides improving mutation effect predictions, QDPR also provides a novel framework for synergistically combining high-throughput simulation data with ML-guided protein engineering, even when the molecular basis of the desired function is not known in advance.

Additionally, because QDPR proceeds by computing a score representing the consensus of a large number of individual models, in principle other advanced ML approaches could be included alongside the dynamic feature prediction models, either as a replacement for or in addition to the direct sequence-to-fitness model already included, or to directly improve the quality of dynamic feature predictions. This is likely to be of significant benefit because the additional information provided by the molecular dynamics simulations is orthogonal – and so probably complementary – to other types of information commonly used to guide variant selection, such as evolutionary context or predicted fold stability. A detailed analysis of this possibility is outside the scope of this work; however, as a first pass, the effect of including zero-shot predictions from ProSST[39] (the current best overall zero-shot model against the ProteinGym benchmarking suite[34]) as a separate feature alongside the QDPR models trained to predict dynamic features was assessed on the full *Av*GFP dataset used in this work. Based on one trial, this increased the spearman rank coefficient for QDPR trained on just eight

labeled sequences from 0.50 to 0.56. The coefficient for ProSST alone was 0.37.

The most straightforward application of the method presented here is to the engineering of proteins where collection of large experimental datasets using methods like deep mutational scanning is not possible or prohibitive, making most supervised ML-guided approaches undesirable due to insufficient training data. More generally, this method will likely be of value in any case where the time and resource commitment associated with performing hundreds-to-thousands of protein simulations is lower than that associated with collecting a large mutation effect dataset. However, even in relatively data-rich cases, this approach might be used to augment the existing dataset to improve predictions and to enhance the interpretability of the results. Simulation data can also be collected in parallel with benchtop data to avoid increasing the overall lifespan of a protein engineering project.

Additionally, as a structure is the only information about the protein required to perform the necessary simulations, this method expands the reach of simulation methods in protein engineering to a broad range of protein targets where the exact relationship between molecular dynamics and function is not known. Indeed, ranked feature importance scores will in such cases prove useful in helping researchers to formulate informed hypotheses of the molecular basis of protein function. This is especially salient in light of the recent successes of protein structure prediction models like AlphaFold[40] and RoseTTAFold[41] in expanding the range of proteins for which high-confidence structure predictions are available.

Furthermore, this method should not be construed as being limited to applications in protein engineering; a similar approach could be used to guide the interpretation of mutation effect datasets by unifying experimental and computational data in order to provide molecular-level interpretations. For example, data describing different phenotypic effects of mutations to proteins *in vivo* could be interpreted in the context of biophysical features from simulations in order to help guide study into the molecular origin of disease. This potential application highlights the implications QDPR has not only for protein engineering, but for protein science in general.

Conclusion

Here is presented a new method for selecting improved protein variants, and for interpreting the molecular-level effects of mutations on arbitrary experimentally determined properties, based on machine learning applied to biophysical features extracted from atomistic molecular dynamics simulations of protein variants. This method has been shown to permit the selection of improved protein variants across two highly distinct proteins and functions using only very small amounts of experimentally labeled protein variant data, whereas ML-guided directed evolution without simulation data is shown to be greatly inferior in such cases. Furthermore, this method has been shown to be capable of accurately predicting the key residues implicated in protein function, despite the fact that the simulations are highly general and do not require prior knowledge of the molecular basis of the target property. This work expands the reach of molecular simulation for protein engineering and protein science more broadly, both in cases where large amounts of training data for traditional ML approaches are available, and where they are not.

Supporting Information Available

The following files are available free of charge.

- Supporting Information: additional data and methodological details
- pearson_scores.xlsx: Pearson's correlation coefficients for each trained biophysical model on the validation datasets
- importance_rankings.xlsx: complete lists of consensus ranked importance for features in GB1 and *Av*GFP in descending order after the first round of each QDPR campaign

Data Availability

All MD inputs, Python scripts for MD analysis and training models, and checkpoint files for models trained on MD data are available at: doi.org/10.5281/zenodo.15232525

Author Contributions

TEB is the sole author of this work and was responsible for project ideation, data collection and analysis, and manuscript preparation.

Competing Interests

The author declares no competing interests.

References

(1) Mazurenko, S.; Prokop, Z.; Damborsky, J. Machine Learning in Enzyme Engineering. *ACS Catalysis* **2020**, *10*, 1210–1223.

(2) 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.

(3) Fowler, D. M.; Fields, S. Deep Mutational Scanning: A New Style of Protein Science. *Nature Methods* **2014**, *11*, 801–807.

(4) Qiu, Y.; Hu, J.; Wei, G.-W. Cluster Learning-Assisted Directed Evolution. *Nature Computational Science* **2021**, *1*, 809–818.

(5) Frisby, T. S.; Langmead, C. J. Bayesian Optimization with Evolutionary and Structure-

Based Regularization for Directed Protein Evolution. *Algorithms for Molecular Biology* **2021**, *16*, 13.

(6) Wittmann, B. J.; Yue, Y.; Arnold, F. H. Informed Training Set Design Enables Efficient Machine Learning-Assisted Directed Protein Evolution. *Cell Systems* **2021**, *12*, 1026–1045.e7.

(7) Zhang, Q. et al. Integrating Protein Language Models and Automatic Biofoundry for Enhanced Protein Evolution. *Nature Communications* **2025**, *16*, 1553.

(8) Biswas, S.; Khimulya, G.; Alley, E. C.; Esvelt, K. M.; Church, G. M. Low-N Protein Engineering with Data-Efficient Deep Learning. *Nature Methods* **2021**, *18*, 389–396.

(9) Meier, J.; Rao, R.; Verkuil, R.; Liu, J.; Sercu, T.; Rives, A. Language Models Enable Zero-Shot Prediction of the Effects of Mutations on Protein Function. Proceedings of the 35th International Conference on Neural Information Processing Systems. Red Hook, NY, USA, 2021; pp 29287–29303.

(10) Luo, Y.; Jiang, G.; Yu, T.; Liu, Y.; Vo, L.; Ding, H.; Su, Y.; Qian, W. W.; Zhao, H.; Peng, J. ECNet Is an Evolutionary Context-Integrated Deep Learning Framework for Protein Engineering. *Nature Communications* **2021**, *12*, 5743.

(11) D. A. Case et al. Amber 2021. University of California, 2021.

(12) Gallagher, T.; Alexander, P.; Bryan, P.; Gilliland, G. L. Two Crystal Structures of the B1 Immunoglobulin-Binding Domain of Streptococcal Protein G and Comparison with NMR. *Biochemistry* **1994**, *33*, 4721–4729.

(13) Shinobu, A.; Palm, G. J.; Schierbeek, A. J.; Agmon, N. Visualizing Proton Antenna in a High-Resolution Green Fluorescent Protein Structure. *Journal of the American Chemical Society* **2010**, *132*, 11093–11102.

(14) Tian, C.; Kasavajhala, K.; Belfon, K. A. A.; Raguette, L.; Huang, H.; Migues, A. N.; Bickel, J.; Wang, Y.; Pincay, J.; Wu, Q.; Simmerling, C. ff19SB: Amino-Acid-Specific Protein Backbone Parameters Trained against Quantum Mechanics Energy Surfaces in Solution. *Journal of Chemical Theory and Computation* **2020**, *16*, 528–552.

(15) Izadi, S.; Onufriev, A. V. Accuracy Limit of Rigid 3-Point Water Models. *The Journal of Chemical Physics* **2016**, *145*, 074501.

(16) Anandakrishnan, R.; Aguilar, B.; Onufriev, A. V. H++ 3.0: Automating pK Prediction and the Preparation of Biomolecular Structures for Atomistic Molecular Modeling and Simulations. *Nucleic Acids Research* **2012**, *40*, 537–541.

(17) Hopkins, C. W.; Le Grand, S.; Walker, R. C.; Roitberg, A. E. Long-Time-Step Molecular Dynamics through Hydrogen Mass Repartitioning. *Journal of Chemical Theory and Computation* **2015**, *11*, 1864–1874.

(18) Chaudhury, S.; Lyskov, S.; Gray, J. J. PyRosetta: A Script-Based Interface for Implementing Molecular Modeling Algorithms Using Rosetta. *Bioinformatics* **2010**, *26*, 689–691.

(19) Olsson, M. H. M.; Søndergaard, C. R.; Rostkowski, M.; Jensen, J. H. PROPKA3: Consistent Treatment of Internal and Surface Residues in Empirical pKa Predictions. *Journal of Chemical Theory and Computation* **2011**, *7*, 525–537.

(20) Roe, D. R.; Cheatham, T. E. I. PTRAJ and CPPTRAJ: Software for Processing and Analysis of Molecular Synamics Trajectory Data. *Journal of Chemical Theory and Computation* **2013**, *9*, 3084–3095.

(21) 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.

(22) Gowers, R.; Linke, M.; Barnoud, J.; Reddy, T.; Melo, M.; Seyler, S.; Domański, J.; Dotson, D.; Buchoux, S.; Kenney, I.; Beckstein, O. MDAnalysis: A Python Package for the Rapid Analysis of Molecular Dynamics Simulations. Python in Science Conference. Austin, Texas, 2016; pp 98–105.

(23) Michaud-Agrawal, N.; Denning, E. J.; Woolf, T. B.; Beckstein, O. MDAnalysis: A Toolkit for the Analysis of Molecular Dynamics Simulations. *Journal of Computational Chemistry* **2011**, *32*, 2319–2327.

(24) Kabsch, W.; Sander, C. Dictionary of Protein Secondary Structure: Pattern Recognition of Hydrogen-Bonded and Geometrical Features. *Biopolymers* **1983**, *22*, 2577–2637.

(25) Wernet, P.; Nordlund, D.; Bergmann, U.; Cavalleri, M.; Odelius, M.; Ogasawara, H.; Näslund, L. Å.; Hirsch, T. K.; Ojamäe, L.; Glatzel, P.; Pettersson, L. G. M.; Nilsson, A. The Structure of the First Coordination Shell in Liquid Water. *Science* **2004**, *304*, 995–999.

(26) Singh, S.; Bowman, G. R. Quantifying Allosteric Communication via Both Concerted Structural Changes and Conformational Disorder with CARDS. *Journal of Chemical Theory and Computation* **2017**, *13*, 1509–1517.

(27) Kawashima, S.; Pokarowski, P.; Pokarowska, M.; Kolinski, A.; Katayama, T.; Kanehisa, M. AAindex: Amino Acid Index Database, Progress Report 2008. *Nucleic Acids Research* **2008**, *36*, D202–205.

(28) Olson, C. A.; Wu, N. C.; Sun, R. A Comprehensive Biophysical Description of Pairwise Epistasis throughout an Entire Protein Domain. *Current Biology* **2014**, *24*, 2643–2651.

(29) Sarkisyan, K. S. et al. Local Fitness Landscape of the Green Fluorescent Protein. *Nature* **2016**, *533*, 397–401.

(30) Sauer-Eriksson, A. E.; Kleywegt, G. J.; Uhlén, M.; Jones, T. A. Crystal Structure of the C2 Fragment of Streptococcal Protein G in Complex with the Fc Domain of Human IgG. *Structure* **1995**, *3*, 265–278.

(31) 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.

(32) Yang, J.; Lal, R. G.; Bowden, J. C.; Astudillo, R.; Hameedi, M. A.; Kaur, S.; Hill, M.; Yue, Y.; Arnold, F. H. Active Learning-Assisted Directed Evolution. *BioRxiv* **2024**,

(33) 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.

(34) Notin, P.; Kollasch, A.; Ritter, D.; van Niekerk, L.; Paul, S.; Spinner, H.; Rollins, N.; Shaw, A.; Orenbuch, R.; Weitzman, R.; Frazer, J.; Dias, M.; Franceschi, D.; Gal, Y.; Marks, D. ProteinGym: Large-scale Benchmarks for Protein Fitness Prediction and Design. *Advances in Neural Information Processing Systems* **2023**, *36*, 64331–64379.

(35) Ruscio, J. Z.; Kohn, J. E.; Ball, K. A.; Head-Gordon, T. The Influence of Protein Dynamics on the Success of Computational Enzyme Design. *Journal of the American Chemical Society* **2009**, *131*, 14111–14115.

(36) Janson, G.; Valdes-Garcia, G.; Heo, L.; Feig, M. Direct Generation of Protein Conformational Ensembles via Machine Learning. *Nature Communications* **2023**, *14*, 774.

(37) Aranganathan, A.; Gu, X.; Wang, D.; Vani, B. P.; Tiwary, P. Modeling Boltzmann-weighted Structural Ensembles of Proteins Using Artificial Intelligence–Based Methods. *Current Opinion in Structural Biology* **2025**, *91*, 103000.

(38) Audagnotto, M.; Czechtizky, W.; De Maria, L.; Käck, H.; Papoian, G.; Tornberg, L.; Tyrchan, C.; Ulander, J. Machine Learning/Molecular Dynamic Protein Structure Prediction Approach to Investigate the Protein Conformational Ensemble. *Scientific Reports* **2022**, *12*, 10018.

(39) Li, M.; Tan, P.; Ma, X.; Zhong, B.; Yu, H.; Zhou, Z.; Ouyang, W.; Zhou, B.; Hong, L.; Tan, Y. ProSST: Protein Language Modeling with Quantized Structure and Disentangled Attention. 2024.

(40) Varadi, M. et al. AlphaFold Protein Structure Database: Massively Expanding the Structural Coverage of Protein-Sequence Space with High-Accuracy Models. *Nucleic Acids Research* **2022**, *50*, D439–D444.

(41) Baek, M. et al. Accurate Prediction of Protein Structures and Interactions Using a Three-Track Neural Network. *Science* **2021**, *373*, 871–876.