ACS Synthetic Biology

pubs.acs.org/synthbio

Research Article

Predicting and Interpreting Protein Developability Via Transfer of Convolutional Sequence Representation

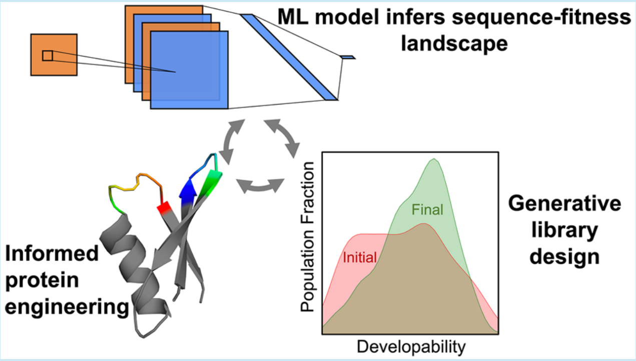
Published as part of the ACS Synthetic Biology virtual special issue "AI for Synthetic Biology".

Alexander W. Golinski, Zachary D. Schmitz, Gregory H. Nielsen, Bryce Johnson, Diya Saha, Sandhya Appiah, Benjamin J. Hackel,* and Stefano Martiniani*

Cite This: *ACS Synth. Biol.* 2023, 12, 2600−2615

Read Online

ACCESS | Metrics & More | Article Recommendations | SI Supporting Information


ML model infers sequence-fitness landscape
Informed protein engineering
Population Fraction
Final
Initial
Developability
Generative library design

ABSTRACT: Engineered proteins have emerged as novel diagnostics, therapeutics, and catalysts. Often, poor protein developability—quantified by expression, solubility, and stability—hinders utility. The ability to predict protein developability from amino acid sequence would reduce the experimental burden when selecting candidates. Recent advances in screening technologies enabled a high-throughput (HT) developability dataset for 10^5 of 10^{20} possible variants of protein ligand scaffold Gp2. In this work, we evaluate the ability of neural networks to learn a developability representation from a HT dataset and transfer this knowledge to predict recombinant expression beyond observed sequences. The model convolves learned amino acid properties to predict expression levels 44% closer to the experimental variance compared to a non-embedded control. Analysis of learned amino acid embeddings highlights the uniqueness of cysteine, the importance of hydrophobicity and charge, and the unimportance of aromaticity, when aiming to improve the developability of small proteins. We identify clusters of similar sequences with increased recombinant expression through nonlinear dimensionality reduction and we explore the inferred expression landscape via nested sampling. The analysis enables the first direct visualization of the fitness landscape and highlights the existence of evolutionary bottlenecks in sequence space giving rise to competing subpopulations of sequences with different developability. The work advances applied protein engineering efforts by predicting and interpreting protein scaffold expression from a limited dataset. Furthermore, our statistical mechanical treatment of the problem advances foundational efforts to characterize the structure of the protein fitness landscape and the amino acid characteristics that influence protein developability.

KEYWORDS: *protein, developability, sequence, landscape, predictive, model*

INTRODUCTION

Engineered proteins have broad utility as therapeutics,[1] diagnostics,[2] targeted drug-delivery vehicles,[3] and as commercial products including industrial enzymes,[4] and agricultural processing catalysts.[5,6] Beyond the primary function (such as binding affinity or enzymatic activity), the utility of the protein is also dependent on the ability to be manufactured, transported, and stored while maintaining functionality. Commonly termed developability,[7,8] this often-overlooked property—quantified by stability, solubility, and production yield—is not typically assessed until late in the commercialization pipeline.[9,10] Late-stage developability failures: (i) require substantial time for engineering or discovery of a new lead, (ii) add avoidable costs which are often passed on to the consumer, and (iii) prevent the immediate use of proteins that would otherwise benefit society.[11] The ability to predict protein developability and design beneficial mutations would ease the manufacturing process by reducing the experimental effort in selecting lead candidates for further evaluation.[11,12]

Predicting protein developability from amino acid sequence is nontrivial due to a myriad of factors: (i) the combinatorial space resulting from 20 canonical amino acids possible at each position produces an exceedingly large sequence domain, (ii) the sequence-developability landscape is believed to be rugged where a single mutation has the ability to eliminate functionality,[13] and (iii) traditional developability assays often drastically undersample the landscape due to exper-

Received: March 31, 2023
Published: August 29, 2023

https://doi.org/10.1021/acssynbio.3c00196

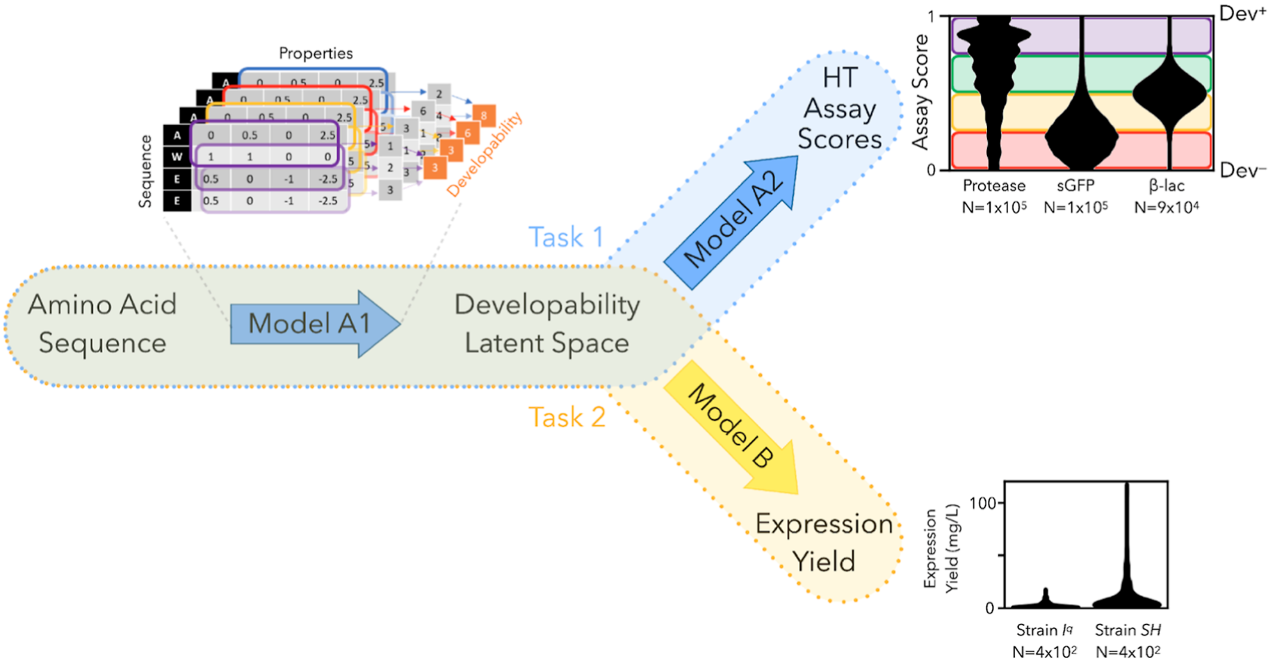


Figure 1. Prediction of protein developability via transfer learning. A sequence-based model to predict developability is trained in two steps. Task 1 (blue, top): large database of protein assay scores is used to train a mapping (model A1) from amino acid sequence to HT assay scores through a learned developability latent space representation (DevRep). Task 2 (yellow, bottom): by transferring the representation, the expression yield (a traditional metric of developability) can be predicted when training a top model with a smaller dataset.

imental constraints.[14] The combination of these factors suggests the creation of a sequence-developability model, and the accurate determination of the most beneficial mutations will require advanced models and sampling techniques.

Recent advances in protein modeling suggest that machine learning possesses the ability to accurately predict functionality with sufficiently thorough and high-quality training data.[15,16] However, it remains unclear which embedding, or numeric representation, for proteins results in the most accurate and efficiently trained model. The traditional one-hot (OH) embedding for categorical variables creates a sparse embedding that lacks knowledge of physicochemical similarities between amino acids and is likely to result in poor performance.[17,18] Alternative approaches attempt to utilize precomputed amino acid properties, such as AAindex[19] or structurally based properties, such as non-polar surface area or contact density,[20] to embed sequences. However, determining the correct set of properties to use can lead to an exhaustive yet still incomplete search. An increasingly popular approach is to utilize an evolutionary-based model trained from homologs.[21−23] Nevertheless, properties that impact natural proteins (likely including primary function, natural mutational rates, and likelihood of experimental sampling) may not be the properties useful for assessing developability. As a result, we believe that the most efficient method of training a sequence-developability model will use more direct experimental developability proxies, collected in high-throughput (HT), that can be transferred to predict traditional developability metrics. Our proposed method thus reflects current developability pipelines[7,24] in applying informed developability metrics to learn a latent developability profile and extends such methods via leveraging this information to predict a given developability metric of interest.

In this study, we aimed to train and test a sequence-based model to predict one metric of developability—recombinant expression—for variants of the protein ligand scaffold Gp2. While specific variants of this 45−49 amino acid protein scaffold have been shown to possess novel binding activity,[25,26] serve as a diagnostic in PET imaging,[27] and inhibit growth of breast cancer cells,[28] many variants still possess poor developability. In a prior study, a series of three HT assays—on-yeast protease resistance, expression as a fusion with a fragment of split green fluorescent protein (GFP), and modular insertion in split β-lactamase—were validated by mutual information and prediction of Gp2 variant yield (mg/L) via bacterial expression in two *E. coli* strains—T7 Express lysY/I^q (I^q) and SHuffle T7 Express lysY (SH).[17] Herein, we assess the ability to first train a sequence-based machine learning model to predict HT assay performance and transfer the developability representation (DevRep) to improve the accuracy in prediction of a traditional developability metric (Figure 1). After building a predictive model, we (i) analyze the learned sequence representation to identify factors driving recombinant expression, (ii) use enhanced sampling techniques to explore and portray the developability landscape and to identify high-yield variants, and (iii) validate the findings experimentally showing that in silico directed evolution can significantly outperform random mutagenesis.

RESULTS

Protein Embeddings Predict HT Assay Developability. A protein's properties are determined by the interaction between amino acids, with various chemical properties, arranged in a three-dimensional structure uniquely determined by its linear amino acid sequence. We constructed models that first learn amino acid properties and then combine them to create an embedding representative of Gp2 paratope variants (Figure 2a). We considered three architectures: (i) flat—where all amino acid properties at all positions interact at once, (ii) recurrent—where amino acid properties are fed one at a

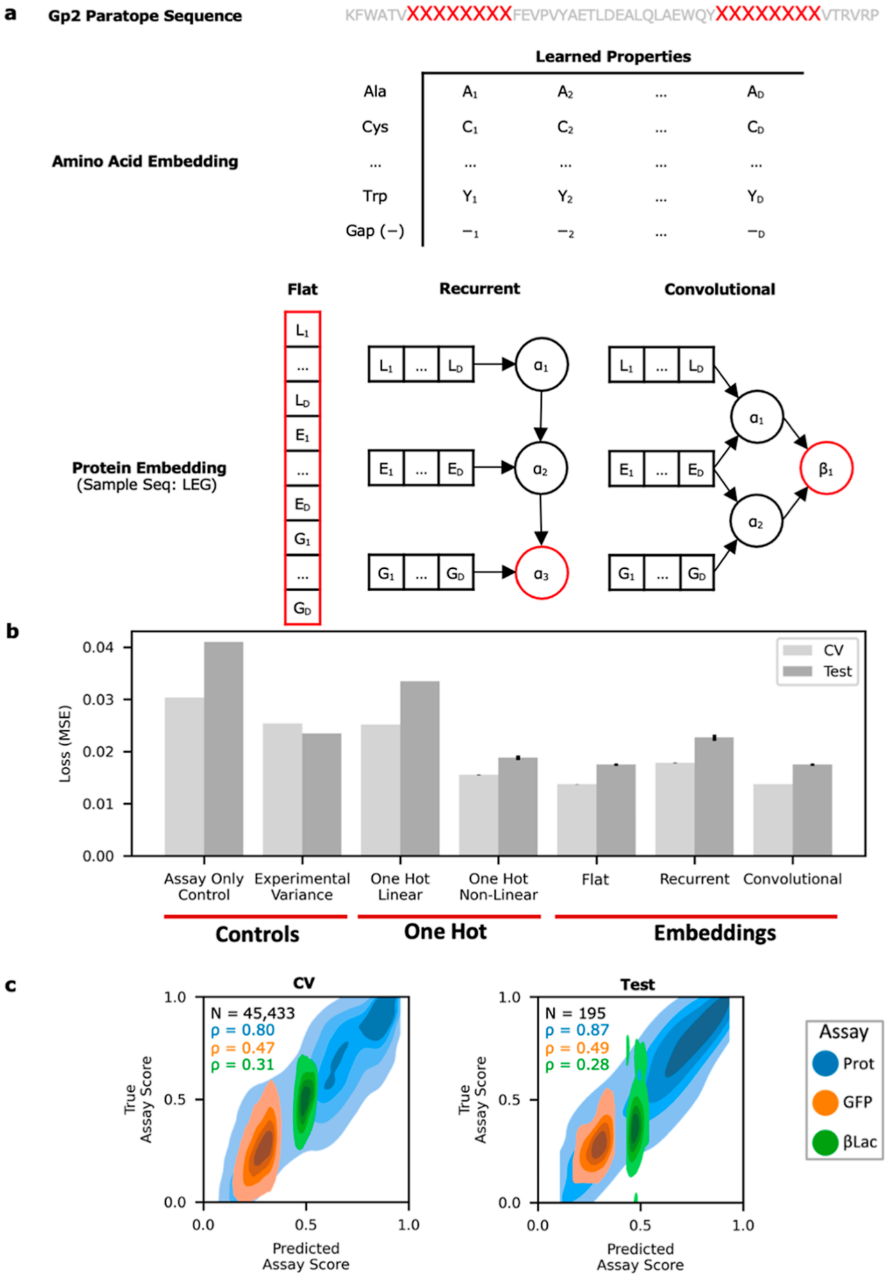


Figure 2. Protein embedding strategies based on interacting amino acid properties predict HT developability assay scores. (a) Gp2 paratope residues are embedded as learned amino acid properties and are combined via three different strategies into a developability representation, identified via a red outline. (b) Embedded and non-embedded (OH) architectures were trained to predict assay scores via CV and evaluated on an independent test set of sequences (independent two-way Student's t-test for embeddings vs non-embeddings $p < 0.05$). (c) Convolutional architecture's predictions are compared to the true assay scores (Prot: protease resistance, GFP: soluble expression in split-GFP system, βLac: modularity in split-β-lactamase) as a kernel density plot. The number of unique Gp2 variants and the Spearman's rank correlation are displayed.

time into a memory unit that is updated as a function of the previously observed positions, and (iii) convolutional—where amino acid properties are first summarized in a local region of the protein and then combined to obtain a full protein embedding. Multitask learning was applied to use all three HT assays to train a single developability embedding. Previous analysis revealed that this set of three HT assays was most informative and least redundant with respect to inferring recombinant yield, a low-throughput traditional metric for developability.[17] We allowed dense layers between the protein embedding and assay scores [one to five layers permitted, hyperparameter optimization during cross-validation (CV) resulted in four layers] after the concatenation of a OH-encoded assay-identifying vector. The range of hyperparameter for the embedding layers and top-model considered during CV are detailed in Tables S1 and S2, respectively; the final set of DevRep hyperparameters is shown in Table S3.

The performance of HT assay score prediction was compared to a series of controls as assessed by the mean-squared error (MSE) of the CV set and an independent test set (Figure 2b). All three architectures using sequence information were more accurate than the assay-only model (independent two-way Student's t-test $p < 0.0001$ for all the three embedding methods). We compared these architectures' performance to

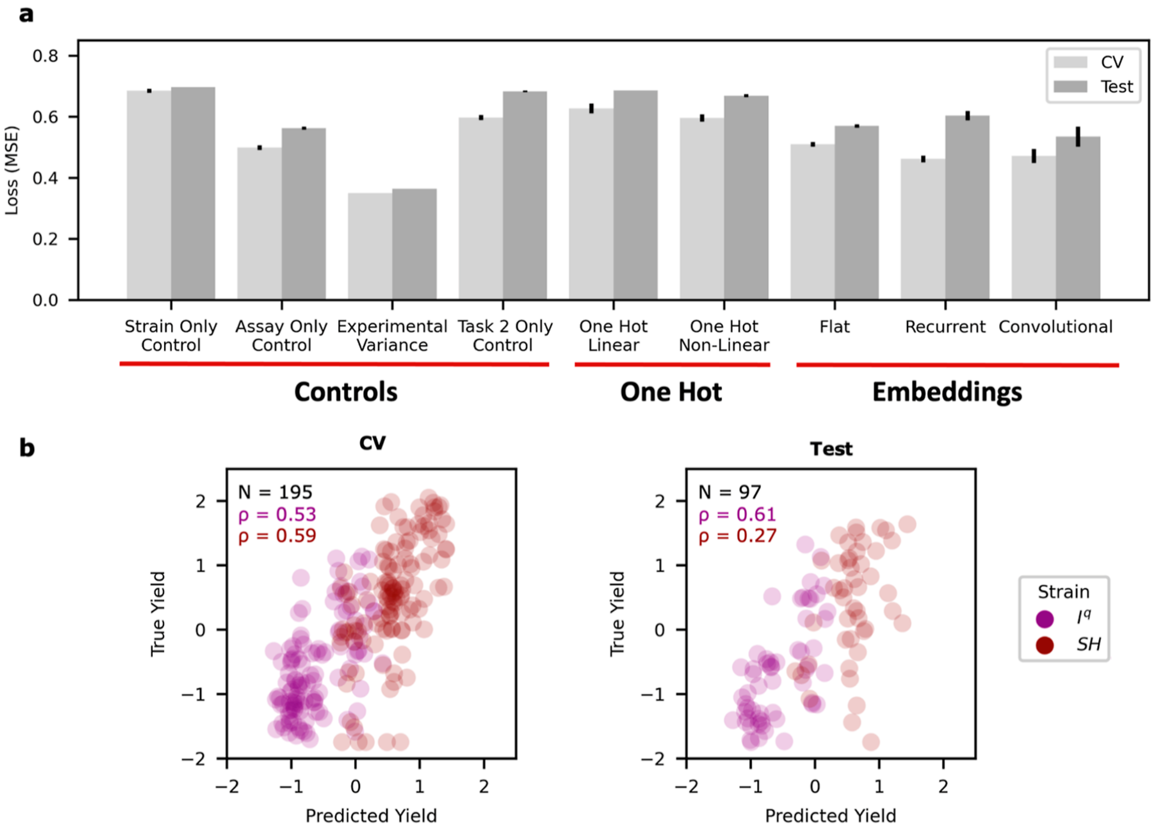


Figure 3. Transferred convolutional embedding predicts yield more accurately than traditional embedding strategy. (a) CV and test performances of predicting yield comparing a traditional OH embedding to protein inspired embeddings trained by HT assay scores. (b) Convolutional embedding with a SVM top model's prediction of yields versus experimentally measured yield across *E. coli* strains I^q and SH.

the experimental MSE, viz. the variance of our measurements. This experimental assay variance was calculated as the sequence-averaged trial-to-trial (N = 3) variance of the assay scores.[17] Dividing the variance by N = 3 yields the squared standard error (SSE) (experimental assay CV SSE: 8.5×10^{-3}; experimental assay test SSE: 7.8×10^{-3}). The SSE represents our confidence in the assay scores when averaged over multiple observations/trials. Interestingly, the protein-inspired architectures can learn from multiple trials and thus predict assay scores with lower MSE than the experimental variance (though not as low as the experimental SSE), highlighting the previously noted low resolution of a single trial of the assays.[17] We also compared the results to models that take as input a flat OH encoding of the amino acids of Gp2 paratope (i.e., without linearly embedding the individual OH vectors first). We observe that a nonlinear model (flat sequence with dense layers between sequence and assay score) significantly outperforms a linear (ridge regression) model (independent two-way Student's t-test $p < 10^{-6}$). The feedforward neural network (FNN) and convolutional neural netwok (CNN) embedding models (which take in linear-embedding amino acid sequences as inputs) in turn significantly outperform the nonlinear model (independent two-way Student's t-test $p < 0.0001$). We then visualized the relative correlation of the convolutional model's predicted versus experimental assay score (Figure 2c). We found that the model was not equally predictive across assays, with the most accurate performance for the on-yeast protease assay.

Testing Transferability to Traditional Developability Metric. Having developed a series of protein embeddings trained on and capable of predicting HT developability assay performance, we asked next if the same embeddings could be transferred to predict a traditional metric of developability. Keeping the embedding parameters constant (Figure 1, model A1), we fit a separate top model (Figure 1, model B) to predict the soluble Gp2 yield in two *E. coli* bacterial strains—I^q and SH—via multitask learning using a OH-encoded strain identifying vector. We used both linear (ridge regression) and nonlinear models [FNN, support vector machine (SVM) and random forest] to account for possible complex interactions between the embeddings and yield.

We found that transferring embeddings trained via assay scores resulted in the prediction of yield more accurately than a model trained directly from both (i) a OH-encoded sequence to yield and (ii) models trained on embeddings directly inferred from task 2 (viz., recombinant yield prediction). During CV, the recurrent embedding with a random forest top model and the convolutional model with an SVM top model exhibited optimal performance (Figure 3a). Upon evaluation of an independent test set, the convolutional embedding with an SVM top model produced the most generalizable model (Figure 3b) while the recurrent embedding suffered from overfitting. Compared to the OH model with a random forest top model, the convolutional embedding reduced the gap to experimental variance (or MSE) by 44%. A Yeo-Johnson transformation was additionally individually applied to these yield measurements to remove correlation between error and yield.[17] The corresponding yield SSE divides the yield experimental variance by N = 3 (experimental yield CV SSE: 0.117; experimental yield test SSE: 0.121). Additionally, the

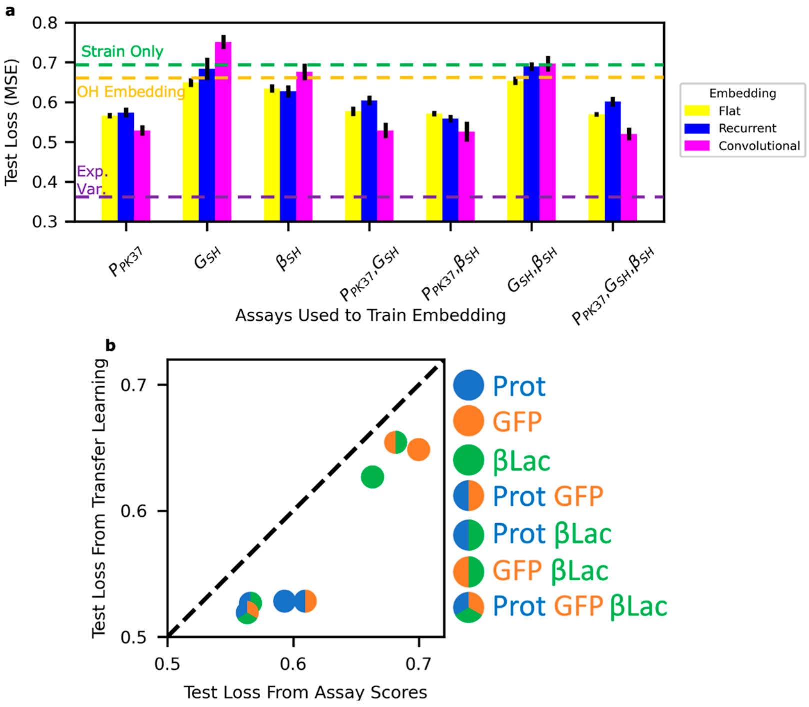


Figure 4. On-yeast protease assay is most informative and transfer learning enables discovery of true signal from imperfect HT assay proxies. (a) Developability representation and top model to yield was trained with combinations of HT assays. The prediction error of sequence yield is grouped by assay combination and colored by embedding architecture. Error bars represent standard deviation of loss from $N = 10$ stochastically trained embeddings and top models. (b) Yield predictions from assay scores and the most accurate trained embeddings for each combination of HT assays suggests that transfer learning is more accurate than models that take as input the experimental assay scores.

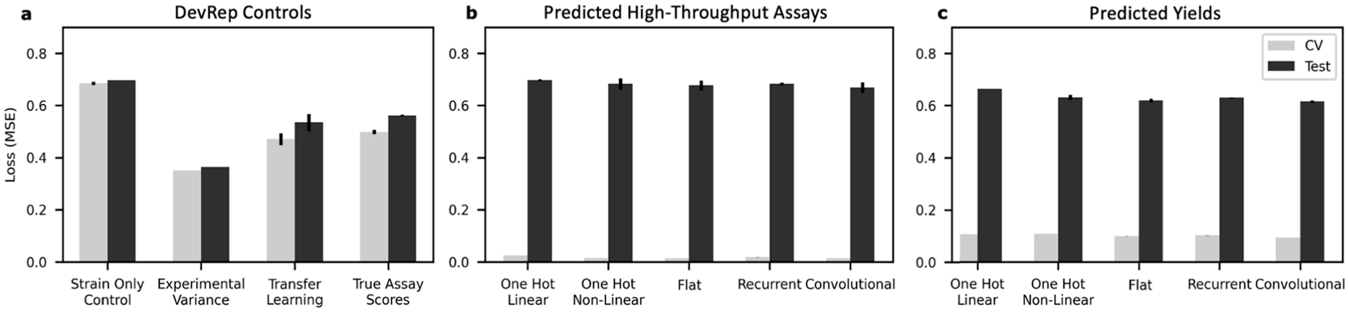


Figure 5. Alternative model CV and test performance. (a) DevRep controls (first outlined in Figure 3a). (b) Predicted high-throughput assays are used to predict yield. (c) Sequence-to-yield model trained on yields predicted from experimental HT data.

convolutional embedding was also able to outperform a model trained on experimentally measured assay scores (CNN test MSE: 0.53 ± 0.03; experimental assay test MSE: 0.56 ± 0.004) ($p < 0.05$, independent two-way Student's t-test) suggesting the embedding can capture experimental assay information at least as well as a more traditional representation of the (experimentally determined) proxy HT assays to yield.

Dependence on HT Assays. Having observed the success of the transfer learning approach utilizing all three HT assays,[17] we sought to (i) understand the importance of each individual assay in creating a transferable embedding and (ii) understand if the transfer learning approach routinely creates a more informative representation than the direct use of HT assay scores. Each combination of HT assays was used to fit the three embedding architectures utilized in this study (flat, recurrent, and convolutional). The three top model architectures (ridge, random forest, and SVM) were first trained on each HT assay combination's embedding. The yield prediction accuracies of these models were then contrasted against those of the optimal top model (Figure 4a). The combination of all three HT assays created the optimal model. Combinations utilizing the on-yeast protease assay resulted in losses lower than those without ($p < 0.01$, independent two-way Student's t-test). In fact, this assay alone achieves an error within 2% of the model utilizing all three HT assays. This suggests that the on-yeast protease assay is the most informative assay and could potentially be used independently in future studies.

The ability of the transfer learning training strategy to identify developability trends and average out noisy signals from similar sequences enables more accurate predictions than direct use of experimental HT assay outputs. Indeed, we previously observed that the transfer learning model slightly outperformed a direct assay score to yield model (Figure 3a). We, therefore, sought to understand if transfer learning was successful because of the use of multiple assays and/or of the learning strategy more generally. To answer this question, we plotted the accuracy of models trained directly on

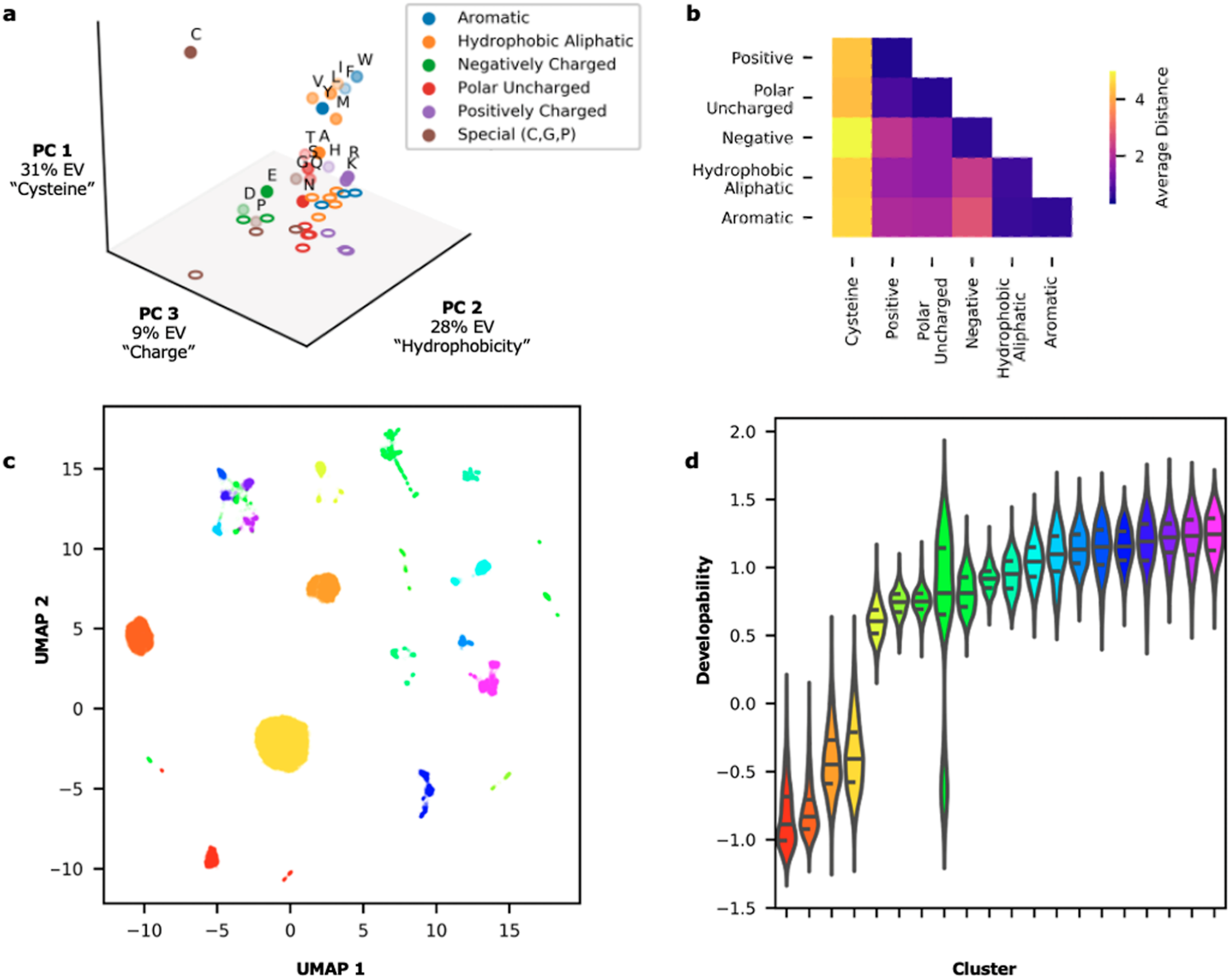


Figure 6. Analysis of trained embeddings reveals properties related to developability. (a) PC of the 19-dimensional amino acid embedding, colored by category of residue. EV = explained variance. (b) Inter- and intraresidue category distances highlighting the uniqueness of cysteine and lack of difference between aromatic and aliphatic residues. (c) Clusters of sequences were identified via UMAP and hdbscan of the 45,433 sequences used for training. (d) Developability, as predicted by yield, varies between clusters trained on HT assay scores.

experimentally measured assay scores to the accuracy of models that utilized the assay scores to train a representation that was transferred to predict yield (Figure 4b). We observed a correlation (Spearman's ρ = 0.96) between the losses, suggesting that the more relevant assay score combinations enable more accurate embeddings regardless of learning strategy. We also observed a significant systematic decrease of loss from transfer learning models compared to models trained directly on assay scores (independent two-way Student's t-test $p < 0.01$), Figure 4b. The ability of transfer learning to consistently outperform assay score models, even when a single assay is used, suggests the model can utilize sequence information to denoise errors present in the assay output.

Alternative Model Building Approaches. After successful construction of the transfer learning model, we sought to validate the optimality of our transfer-learning strategy. In one alternative strategy, we evaluated the possibility of using DevRep to predict HT assay scores from sequences and then to predict the experimental yield measurements from these predicted assay scores (rather than from the DevRep embedded sequences). In another, we evaluated the possibility of predicting yields from the experimental assay scores first and then to train a sequence-to-yield model on these predicted yields (skipping the transfer learning altogether). We display the results of this analysis in Figure 5, showing DevRep ("transfer learning") and relevant controls (a), predicted HT assays (b), and predicted yields (c).

Briefly, both alternative strategies displayed significantly poorer performance and greater overfitting relative to the DevRep controls (Figure 5b,c). Further discussion of these regimes is available in Figure S1 and corresponding sections of Supporting Information and Methods 1 and 1.4, respectively. This analysis supports our choice of developing a transfer learning approach in which only experimental (rather than predicted) assays and yields are used to learn a developability representation that when used as the input (viz., transferred) to a specialized model outperforms alternative approaches.

Dependence on Sample Size. We next sought to understand the relationship between the size of the training datasets and the accuracy of the model. To this end, we randomly subsampled unique sequences from the HT assay

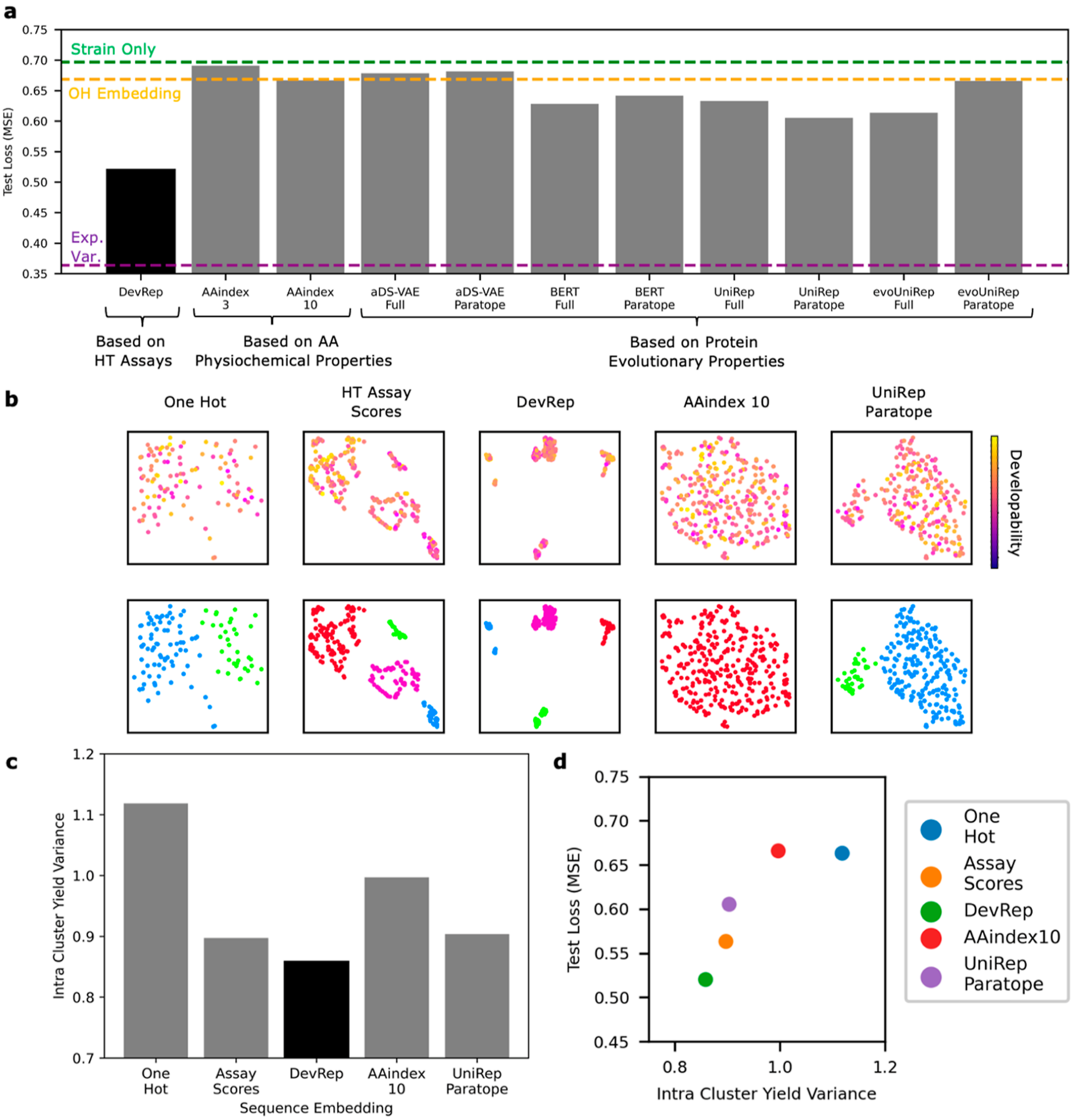


Figure 7. HT assay-trained embedding contains more developability information than alternative embeddings. (a) Comparison of protein representations' ability to predict the yield as represented by the loss on an independent set of sequences. (b) Variants were plotted using UMAP for each embedding. (top) Color represents experimentally measured developability. (bottom) Sequences were clustered by UMAP coordinates. Color represents unique clusters. (c) Variance in predicted yield across sequences within a given cluster. (d) Correlation between the intracluster yield variance and the corresponding models' (trained using the same embedding) predictive performance confirms that models that cluster sequences with similar yield also achieve better predictive performance, indicating that the embedding is informative about the predicted quantity (yield).

dataset to develop convolutional embeddings and compare performance to nontransfer learning models that take as input a simple OH encoded representation of the sequence, or the experimental assay scores (Figure S2). DevRep performance systematically improved relative to controls as training data increased, highlighting further potential performance improvements from more data. The control models show greater efficiency on smaller sample sizes compared with DevRep.

Model Interpretation. We have shown that an accurate model for the prediction of developability metrics such as the soluble yield of Gp2 in *E. coli* can be obtained by transfer learning. Specifically, we utilize a convolutional model to infer an embedding from a set of three HT assays and feed (transfer) the embedding to an SVM (top) model to predict yield. We refer to this model as "DevRep", and in what follows we explore the physical significance of the learned representation and ascend the resulting developability landscape, yielding both a quantitative visualization of the landscape and a library of diverse and highly developable variants. The candidates of this library were then experimentally found to be produced in higher yield than sequences obtained by random mutagenesis.

AA Embedding. First, we analyzed the trained amino acid embeddings to determine what properties are most relevant to the developability of Gp2 (Figure 6a). We evaluated if we could identify linearly separable latent amino acid features from our model via principal component analysis (PCA). The 19 feature dimensions (or inferred "properties") were distilled down to three principal components (PCs) which explain 68% of the total variance. Upon inspection, we determined that cysteine is uniquely separated in PC 1 and 2. Additionally, PC 2 appears to separate the remaining residues by hydrophobicity by placing aromatic and aliphatic residues away from polar and charged residues. PC 3 further separates hydrophilic residues into negative, neutral, and positively charged. Interestingly, histidine (which possesses a pK_a near experimental conditions) is located closer to neutral amino acids compared to arginine (R) and lysine (K), commenting on the ability of the model to learn about both charged states. We then compared each PC to the AAindex[19] list of properties in an attempt to find the most correlative physicochemical property: PC 1—coefficient over single-domain globular proteins ($\rho = 0.91$), a measurement of hydrophobicity[29] again underscoring its importance on developability; PC 2—normalized frequency of N-terminal nonbeta region ($\rho = 0.86$), a measurement of residue frequency in nonstructured regions;[30] and PC3—helix termination parameter at positions j-2, j-1, and j ($\rho = 0.83$), a measurement of residue frequency in short helical structures.[31] Together, PC 2 and 3 suggest that the paratopes may be balancing between a flexible loop and a short helical conformation to provide stability.

We further evaluated the average inter- and intraresidue PC distances (Figure 6b). Each identified cluster of residues has a lower intraresidue distance than inter-residue distance except for aromatic (F, W, Y) and hydrophobic aliphatic (A, I, L, M, V) residues suggesting the hydrophobic nature of these residues outweighed the relative size difference and additional interaction capabilities of aromatic rings.

Clustering of Training Sequences in Developability Space. We next assessed the interaction of the residue embeddings by converting the 97-dimensional DevRep embedding for the 45,433 training sequences via UMAP.[32] UMAP accounts for nonlinearly related features; we thus chose UMAP (rather than PCA) for this analysis as we hypothesize that sequence embeddings are not linearly separable. We then utilized hdbscan[33] to identify 19 clusters of sequences from the two-dimensional UMAP space (Figure 6c). We discovered that the clusters contain information about the variant developability by finding a significant difference in developability distributions as a function of cluster (Figure 6d, Kruskal—Wallis H-test, $p < 0.05$). We further investigated the UMAP distributions of these clusters (Figure S3). This analysis suggests that poorly developable clusters (Figure 6d) are separated along a nonlinear manifold in DevRep space from highly developable clusters (Figure S3a). Additionally, the differences in amino acid frequencies between the selected clusters, paired with the amino acid embedding analysis, suggest that the model learned residues' interactions, particularly with cysteines (Figure S3b).

Comparison to Alternative Protein Embeddings. We next compared DevRep to models built with other state-of-the-art protein embeddings. The AAindex[19] was used to create an embedding based upon physiochemical properties. As the index is known to contain several similar entries, PCA was used to isolate 3 and 10 residue properties. The paratope was then converted into either of these sets of AAindex properties and flattened. We also compared DevRep to four representations trained on evolutionary properties: concatenated OneHot encoding with the evidence lower bound[34] (ELBO)[35] inferred from a DeepSequence variational autoencoder (DS-VAE)[34] trained on HMMER[36] that suggested homologous sequences to Gp2 within the UniRef100[37] database (viz. aDS-VAE), BERT[38] transformer embedding[22] that was trained on the Pfam[39] database via predicting masked residues (viz., BERT), UniRep[21] that was trained autoregressively on the UniRef[37] database (viz., UniRep), and evolutionarily (evo) tuned UniRep (viz., evoUniRep) obtained by isolating homologous sequences to Gp2 via HMMER[40] and updating the embedding via the Jax-UniRep[41] software package. The augmented VAE concatenations were constructed by concatenating ELBOs to OneHot encodings of either the full sequence ("full") or the paratope sequence ("paratope"). Similarly, the BERT and UniRep evolutionary embeddings were tested by averaging over either the full or the paratope sequences.

Each model built on these embeddings was trained to predict yield utilizing the same architectures and hyperparameter search strategy as for DevRep (Figure 7a). We found that DevRep was able to predict yield significantly more accurately than every other embedding. As expected, the evolutionary-based embeddings (particularly UniRep paratope) were able to predict the yield more accurately than the strain-only and OH controls. To aid interpretability of what physicochemical properties enabled this superior performance relative to our controls, we repeated the analysis as shown in Figure 6a,b across the best-performing benchmark embeddings (Figure S4).

DevRep demonstrated the tightest intragroup clustering (0.34 ± 0.18 PC distance) and highest intergroup distinction (2.7 ± 1.4 PC distance), including substantial differentiation of cysteine (4.6 ± 0.2), with only UniRep paratope displaying comparable cysteine component distance (3.7 ± 0.2) with respect to all other physicochemical components as referenced against the top-performing AAindex embedding, AAindex10 (1.7 ± 0.4). This realization demonstrates DevRep's uniquely efficient strength to find and use underlying developability relationships compared to other embedding controls. The poor performance of the AAindex also suggests that traditional physicochemical properties are not the best way to describe Gp2 variant developability.

To ensure that the better performance of DevRep in predicting yield was not due to poor model development, we assessed the relationship between variation of sequences using each embedding and the measured developability. The 195 unique sequences with experimentally measured yield were embedded and transformed for visualization via UMAP (Figure 7b). We performed clustering in the UMAP space and calculated the average intracluster variance of the yield to estimate how much information about developability was encoded in the embedding; we associate lower transformed embedding intracluster variance with richer developability information content and presentation (Figure 7c). We found

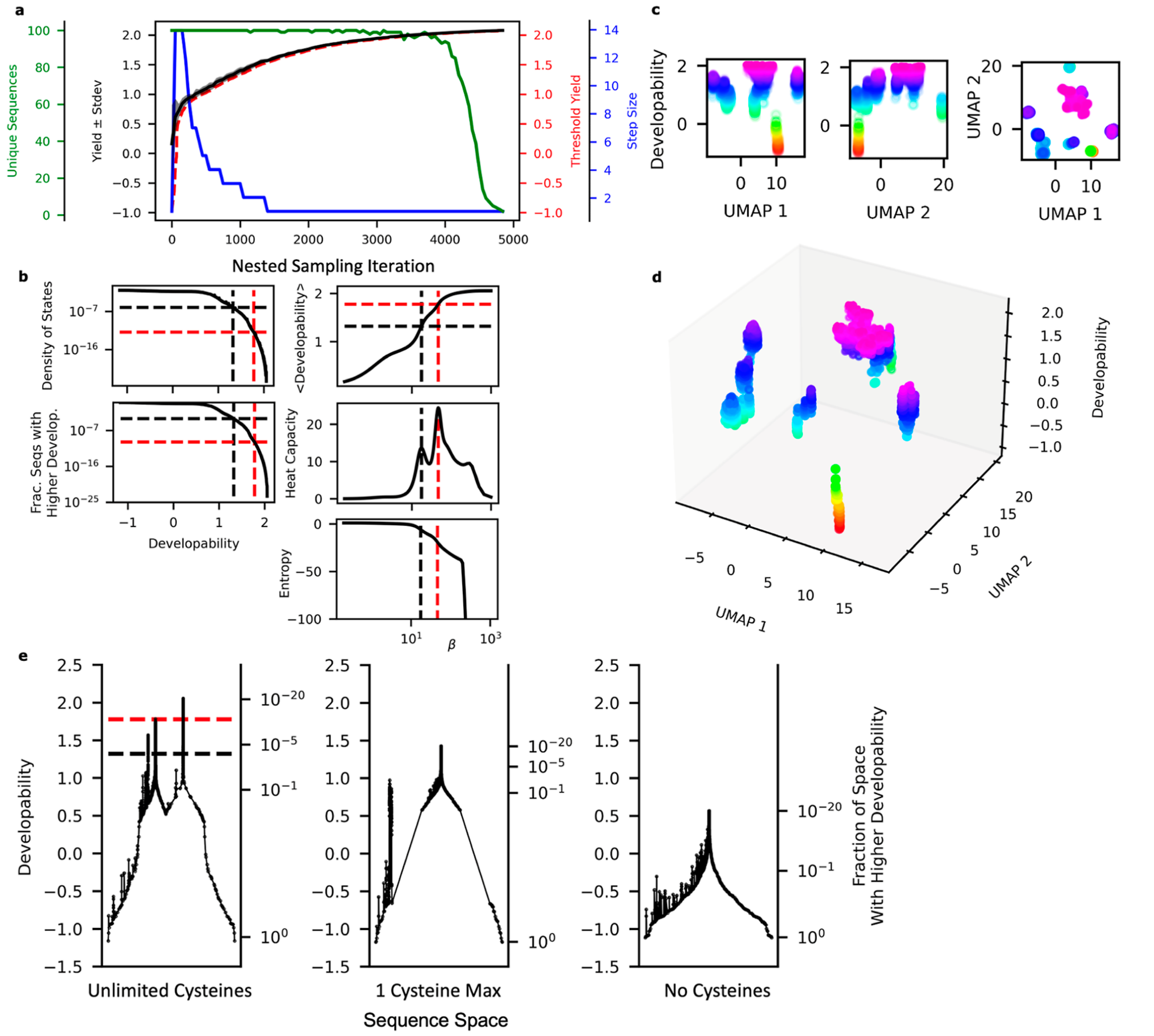


Figure 8. NS characterizes the developability-sequence landscape. (a) NS was performed using 100 evolving sequences while accepting mutations with yields above the threshold per iteration. The threshold yield and corresponding sequences were determined by the lowest yield of the evolving sequences. (b) DOSs for each level of developability were determined and used to estimate the expected developability, heat capacity, and entropy at various inverse temperatures (selective pressure in this context). Two main phase transitions are identified with a dashed line (c,d). The UMAP representation displays the landscape splitting into distinct clusters of DevRep space above the transition. Recorded sequences' predicted developabilities increase from red to purple. (e) Disconnectivity plot for the sequence space displays a landscape with competing developability peaks (when β grows large enough that a lower peak becomes depleted and a higher one enriched, we observe a phase transition).

that the HT assay scores and DevRep's UMAP representation were most informative about yield (Figure 7d).

Sequence Space Analysis Via Nested Sampling. Rather than relying on the skewed experimentally observed distribution of developability, we sought to use nested sampling (NS) to systematically characterize the structure of the fitness landscape while identifying highly developable sequences. At every iteration, NS reduces the fraction of available sequence space "volume" (viz., the number of available sequences) by a constant proportion. Note that this sequence space is analogous to the more general phase space in statistical mechanics. As a result, we can use the output of NS (a list of threshold sequences and their associated yield) to compute the density of states (DOS) as a function of developability (yield). Put simply, we can estimate the relative number of sequences available at any given developability (more generally any quantifiable fitness metric). Computation of the DOS also allows us to determine the analogs of thermodynamic properties such as entropy, mean developability, and developability fluctuations (analogous to the heat capacity computed from the fluctuations in internal energy for a thermodynamic system). For example, in the context of developability, the analog heat capacity measures the rate of change in the mean fitness of the population upon varying selective pressure, β. These thermodynamic analogs help to identify the occurrence of "phase transitions." Such transitions occur when there are competing subpopulations of sequences with different developability, one of which becomes dominant

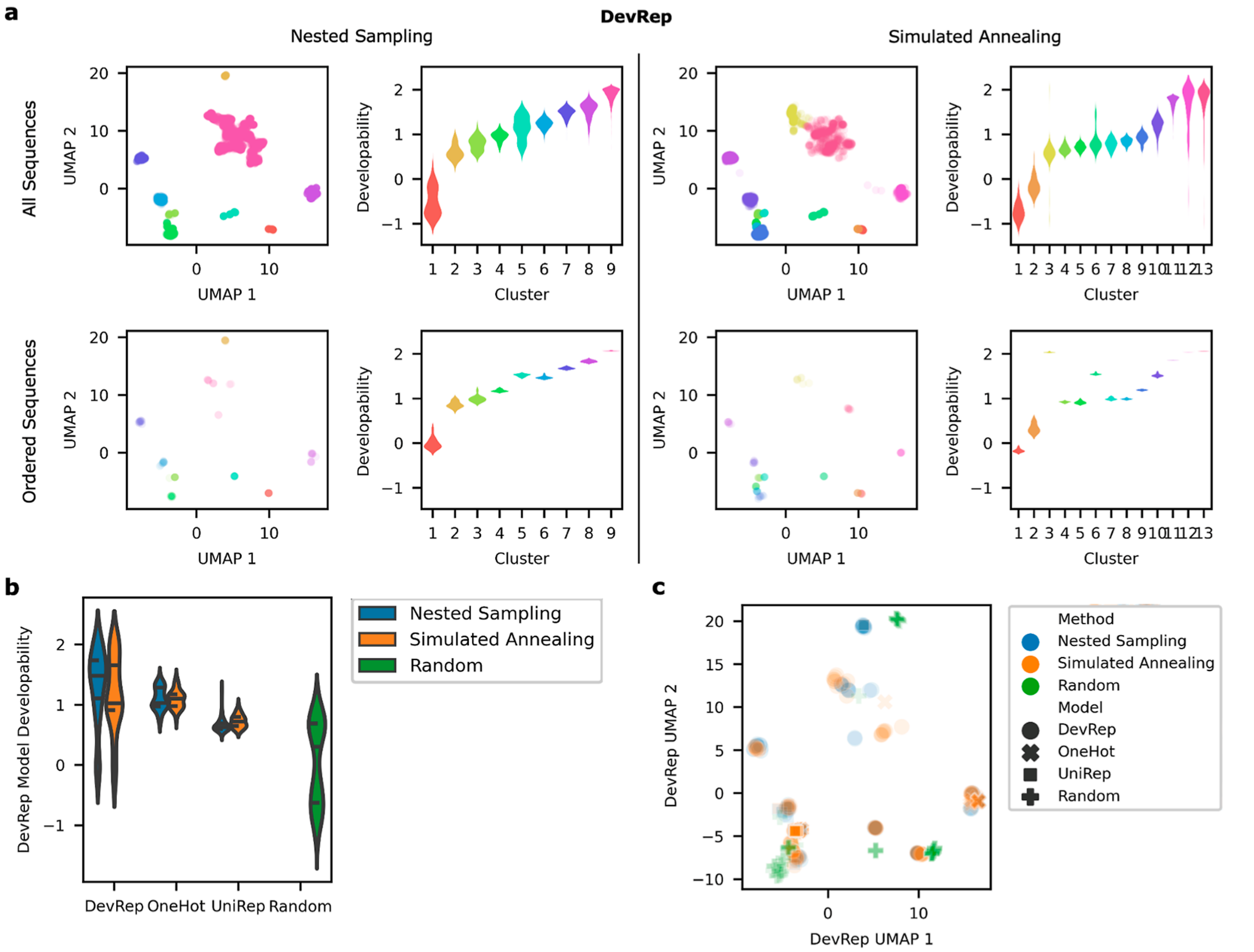


Figure 9. Assessment of DevRep suggested high developability variants. (a) Sequence embeddings identified through either NS (left) or SA (right) strategies were clustered via UMAP (top) (note: we only show the DevRep embedding here). The highest predicted yield variants in each cluster were equally sampled to determine 100 sequences. These variants represent a diverse set of sequences for experimental testing (bottom). (b) Predicted developability distributions according to DevRep using equal intercluster sampling techniques across the sequence variants using different embeddings as in (a). (c) UMAP visualization of top developability variants according to DevRep. Note that the UMAP visualizations of suggested top developability variants for NS and SA in (a) are shown as aggregate in (c).

upon varying selective pressure.[42,43] We ran the algorithm with 100 evolving sequences, removing the lowest yield sequence and thus contracting the phase volume by a factor of 100/101 (~0.99% of its original volume) at every iteration until convergence to a single sequence (Figure 8a). We then utilized the DOS to identify two main transitions: these transitions are apparent in the onset of relatively concave breakpoints corresponding to collapses in the DOS. Between these phase transition regimes, sequences split into multiple competing subpopulations at a given critical selective pressure β (an inverse temperature in a thermodynamic context). These transitions are highlighted by peaks in the heat capacity; the peaks correspond to high variance in sequence space as Gp2 variants transition from one basin of sequence space to another. The expected values of (β, developability) corresponding to these critical temperatures are (17.6, 1.3) and (46.5, 1.8), respectively (Figure 8b). Note that these developability predictions are Yeo-Johnson transformed; these two developabilities, thus, correspond to 4.6 and 13.1 mg/L, respectively. This transition occurs with only 9.3×10^{-5} and 1.3×10^{-10} of all sequences predicted to have a higher yield, respectively.

The output of NS can also be used to visualize the phase space.[44,45] Plotting the sequences in UMAP space shows a single stalk of low developability sequences up to the first phase transition where several high developability clusters exist (Figure 8c,d). The split suggests that beyond the first phase transition, there exists several distinct modes of achieving high developability. A disconnectivity plot of the sequence space was synthesized by creating a graph of nearest sequences of higher yield in the DevRep embedding[43−45] (Figure 8e). The phase transition at the noted critical developabilities corresponds to a sharp decrease in available sequence space and branching occurs when subgraphs of sequences become disconnected at a critical developability level.

We compared disconnectivity plots and UMAP landscapes of the OneHot and UniRep paratope models' embeddings (Figure S5). Every model suggests at least one steep contraction of configuration space corresponding to a basin of similar sequences with high developability. The DevRep landscape is the only embedding to show a large split of sequence space into two competing basins. The OneHot UMAP landscape appears to have sequences of various predicted developability located at every UMAP location,

https://doi.org/10.1021/acssynbio.3c00196
ACS Synth. Biol. 2023, 12, 2600−2615

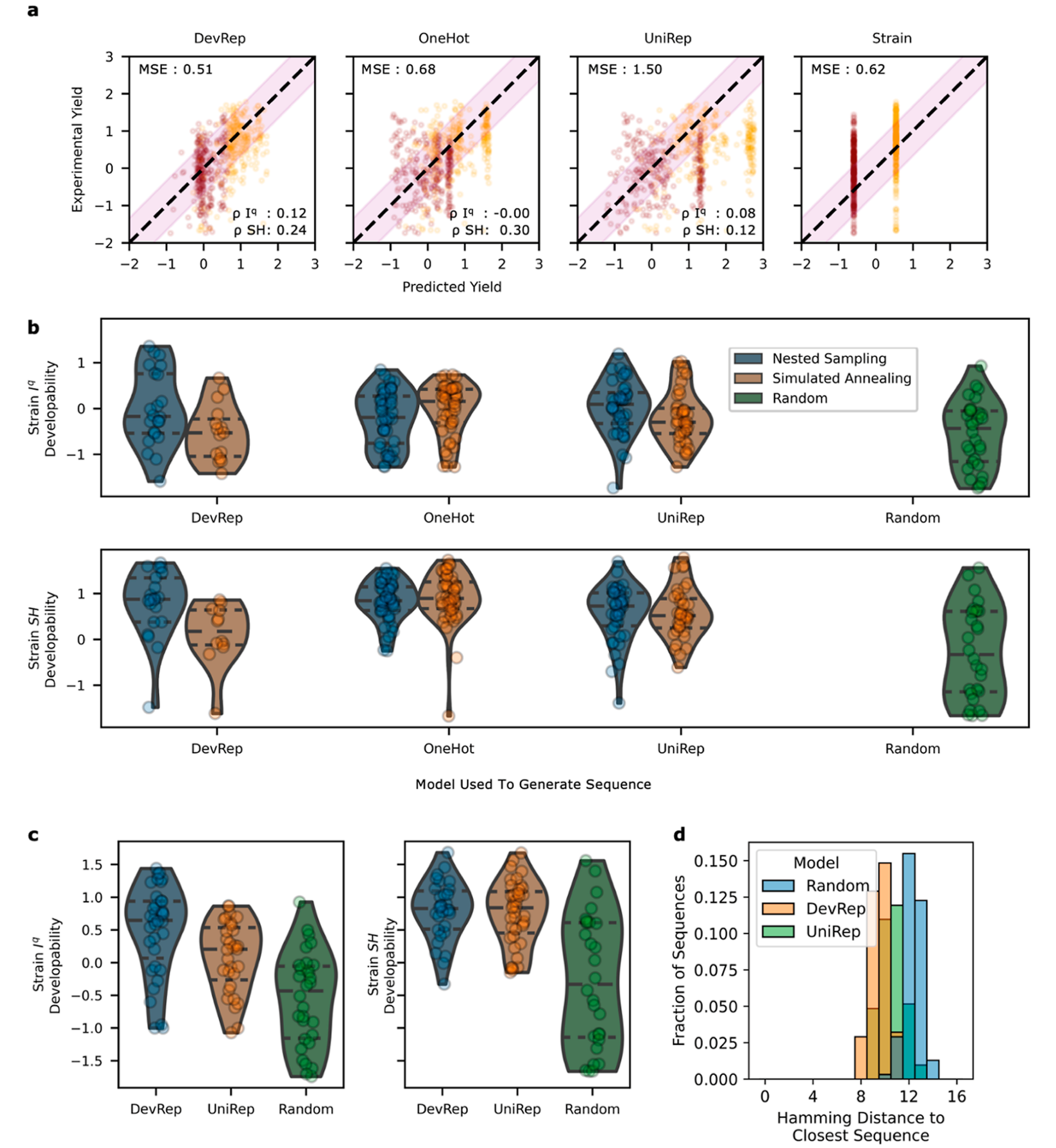


Figure 10. DevRep enables design of developable protein variants. (a) The predicted versus actual developability of 280 I^q and 269 SH variants identified via sampling strategies (see Figures 9, S6 and S7). (b) Sequences generated by each embedding and sampling strategy are compared to each other and to a selection of randomly generated sequences. (c) An additional set of sequences identified via NS of DevRep and UniRep were also compared. These sequences were designed to be more developable and more similar in embedding space. (d) Each sequence in (c) was compared to the set of sequences with measured yield that was used during model training. The distribution shown is broken down by the model used to generate the sequences.

confirming the OH embedding lacks easily interpretable developability information. The UniRep paratope landscape does show correlation between UMAP 1 and developability, suggesting that there is some shared information between UniRep's embedding space and developability space.

Previously, we observed the importance of cysteine in distinguishing developability sequences according to DevRep's embeddings (Figure 6). We hypothesized that restricting cysteine mutations within NS would dramatically influence the resulting sequence-developability landscape. We thus further compared disconnectivity plots and UMAP landscapes of

DevRep models' embeddings when sampled sequences allowed for either (i) at most one cysteine within a sequence or (ii) no cysteine at all (Figure 8e). Indeed, we observe both the disappearance of the second basin within the disconnectivity plots and a significantly less developable final optimal sequence as cysteine is more stringently restricted.

Generation and Experimental Validation of Top Developability Variants. As a final test of the transfer model approach to predict protein developability, we sought to measure the ability to predict high developability variants. Because we found that the Gp2 library splits into many subgroups of sequences that can achieve high developability (as it is also clearly visible from the multiple basins with high developability in the corresponding disconnectivity graph, see Figure 8e), we generated diverse sequences. We also identified sequences using simulated annealing (SA)[46] to compare sampling strategies. The embeddings from each sampling approach were reduced via UMAP and clustered via hdbscan to identify sequences from clusters that are diverse in DevRep space. We then equally sampled across each cluster to acquire diverse high-yield variants. The most developable variants equally sampled in each cluster were recorded to yield a total of 100 final variants (Figures 9a and S6).

The same process was repeated using different embeddings (OneHot and UniRep) for the paratope model. A randomly generated set of sequences were also tested for comparison. The predicted yields and different locations within sequence space for the isolated sequence embeddings suggest that each model has its own distinct maximum and underlying approximation of developability space (Figures 9a and S6).

It was observed that including sequence diversity in the selection scheme introduced lower developability variants. Additionally, large clusters of high developability sequences were observed in both DevRep and UniRep embeddings from NS (Figure S6). Thus, for each model, the large high developability cluster was split into subclusters where 100 additional variants were experimentally evaluated equally spread across the high developability subclusters (Figure 10). In total, 600 variants were thus proposed for experimental characterization.

We measured the expression yield from the soluble fraction of *E. coli* bacteria for 280 variants in the I^q strain and 269 variants in the SH strain. The DevRep model was the most accurate in prediction of unseen sequences (Figure 10a). Interestingly, both our control models outperformed the UniRep-encoded model (Figure 10a) despite UniRep's test MSE beating that of our controls (Figure 7a). We believe that this discrepancy could arise from UniRep having low test error on low-to-medium developability of Gp2 candidates but significantly fail to generalize highly developable candidates relative to the spread of error of our controls.

We next assessed which model and sampling technique identified the top-performing variants with additional focus on diversity (Figure 10b). DevRep identified the highest yield sequences (upper quartile developability: 0.76 for DevRep vs 0.27, 0.35, and −0.05 for OneHot, UniRep, and random, respectively, in strain I^q and 1.34 vs 1.15, 1.01, and 0.61 in strain SH; Figure 10b). NS was more effective than SA for DevRep and UniRep but not OneHot embeddings in both strains.

We then assessed the distribution of yields obtained with a higher focus on developability than diversity (see Figure S7). Again, both DevRep and UniRep embeddings were able to select sequences with higher developability than a random selection (Figure 10c). Additionally, DevRep was able to identify the sequence with the highest developability in I^q. In aggregate, we note the significant difference between the distribution of high developability Gp2 variants between our initial dataset used to train/test DevRep and the final generatively suggested experimentally validated 549 Gp2 variants (Figure S8) to guide our Gp2 libraries toward higher developability. Of final note, we found the sequences identified in this final evaluation were significantly far (in terms of Hamming distance) from variants evaluated during model training. DevRep's sequences were 9.5 (on average) amino acid mutations away from the closest sequence during training (Figure 10d).

These results display a promising utility of DevRep in terms of both predictive accuracy and the ability to identify highly developable variants over current state-of-the-art universal sequence embedding techniques. Note that DevRep is specific to modeling developability of a 12−16 residue paratope (viz., 20^{16} possible variants) of 45,433 sequences, whereas these universal sequence embedding techniques were originally trained on up to only 3 orders of magnitude more sequences to represent all possible sequences found in nature (i.e., $x \gg 20^{16}$ possible sequences). That is, DevRep is significantly more economical in its usage of protein variants relative to these large universal models. Therefore, it is not surprising that DevRep outperforms these universal models on our key task of interest: predicting protein developability. Additionally, the performance of both OneHot and UniRep embeddings and sampling strategies suggests that these techniques could be a useful first step in sequence identification, even prior to experimentation. Finally, we found that the combination of machine learning models with NS constitutes a promising strategy for efficient and interpretable in silico directed evolution of proteins.

While the current study only directly assesses Gp2, the approach is readily transferable to other proteins given the efficiency of the HT experimental assays and the availability of the model code. As such, the approach is not limited by protein or metric, although its performance has only been demonstrated with Gp2 on the metric of recombinant yield.

We envision that the approach can fill an important void in developability pipelines. One set of current approaches seeks to establish measures of protein "druglikeness" analogous to Lipinski's "rule of 5"[47] for small molecules.[7,24] In these methods, several developability metrics are measured either experimentally[7] or in silico[24] across a set of clinically relevant proteins. Candidate proteins are then compared against these underlying distributions with scores that fall within heuristic cutoffs begetting acceptance for further investigation. The experimental approaches provide physical interpretability but are currently low-throughput and preclude sequence-based design, whereas the current computational methods are rapid but would benefit from more robust performance. We demonstrate transfer learning of focused HT experiments as an efficient route to predictability and design.

Our outlined approach is generally applicable to protein sequence-function datasets that comprise library-scale HT proxy data on the order of 10^5 protein variants and gold standard assay measurements (e.g., recombinant protein yield) on the order of 10^2 that correlate well with the proxy assay. Deep mutational scanning (DMS) datasets[48−50] can readily serve as suitable HT proxy datasets. We suggest that these extant DMS datasets can be quickly augmented via random sampling and testing of protein variants using classical gold standard assay measurements; which could be completed efficiently. These augmented DMS datasets would be highly amenable to analysis via our approach.

■ CONCLUSION

This work evaluated the ability of using HT developability assays (proxies for a traditional metric) to learn an embedding that is transferable to a predictive model of a traditional metric (e.g., yield) for which only few data points are available (an example of few-shot learning). We determined that this strategy can overcome noise in the proxy assays and achieve significantly better performance than alternatives. We then analyzed the model's predictions to identify unique modes of achieving high developability based upon the location of cysteine and the importance of hydrophobicity and charge. The configuration space was explored via NS which identified a range of developabilities where the sequences are highly clustered and unique, suggesting that a series of sub-libraries may outperform a single design. The transfer learning approach outperforms models based on physiochemical or evolutionary properties, thereby confirming that developability is a complex and unique property and providing a combined experimental/bioinformatics means to integrate developability design into protein discovery and engineering.

■ METHODS

The following section contains a summary of relevant information to perform the training, CV, and testing regimes of DevRep and assessed benchmark embeddings, and statistical mechanics generative modeling (sampling) protocols for fitness landscape construction. Additional methods can be found in the Supporting Information.

Hardware. Training, CV, testing, and generative modeling (NS/SA) analysis was conducted using NVIDIA Tesla K40 GPUs provided by the Minnesota Supercomputing Institute (MSI). For model development, a single K40 core was assigned to each embedding model-top model pair for each assessed task of first (i) HT assay prediction and then (ii) protein recombinant yield prediction. For generative variant modeling and analysis, a single K40 core was assigned to each of 100 evolving sequence walkers to generate and examine synthetic Gp2 mutants in parallel via a shared memory scheme.[51]

Training Setup. All model development was conducted using TensorFlow v2,[43,52] scikit-learn v0.24.1,[53] NumPy v1.21,[54] and Python v3.7.[55] All sets of examined architectures were trained using the Adam optimizer[56] with learning rate 10^{-3}, $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\varepsilon = 10^{-7}$. All to-be-trained parameters over all layers for each examined architecture were initialized via the Glorot Uniform scheme.[57]

For DevRep and our other in-house models, combinations of embedding architectures (Table S1) and all top-models (Table S2) were trained, validated, and tested on appropriate splits of the 45,433 Gp2 variants following Supporting Information Section 1.1 to simultaneously first (i) predict the three optimal developability assays determined in previous work[17] and then (ii) predict protein recombinant yield.

For benchmark embedding models such as UniRep[21] and BERT,[22,38] all benchmark embedding-top model architecture pairs were fit only on our second and final task of protein recombinant yield prediction. This focus on the second task was conducted in the spirit of desiring to compare state-of-the art universal protein embeddings on our primary task of interest (yield prediction) versus that of our in-house transfer learning approach. Additionally, UniRep full and paratope embedding tuning was conducted in an evolutionary evoUniRep scheme.[21,41] Benchmark embedding sizes are listed in Supporting Information Table S8.

Training Protocol. To train, we use Gp2 variants with varying degrees of assay and protein recombinant yield label information as determined from previous work[17] and extensively explained in Supporting Information Section 1. Each set of assay and yield labels were normalized via a Yeo-Johnson power transformation.[58] For DevRep and our other in-house models, all Gp2 sequences were first either OH or ordinally encoded. Only the Gp2 paratope of length 12−16 amino acids was encoded; each paratope was always padded with gap characters in positions 4,5 and 12,13 as needed to obtain a 21 (20 canonical amino acids plus a gap character) × 16 (positions) input matrix (for OH) or a 1 × 16 input vector (for ordinal). Each Gp2 sequence's strain (i.e., I^q [1,0] or SH [0,1]) is distinguished as an additional concatenated 1 × 2 vector (OH) or 1 × 1 entry (ordinal) to the sequence representation.

All benchmark embedding models such as UniRep[41] and BERT[22] already have full protein embeddings. Thus, either the 45−49 full Gp2 variant sequence or 12−16 paratope Gp2 sequence were embedded for each examined benchmark to produce a set of embeddings over the training set of consistent input dimension size. For the augmented DeepSequence VAE embeddings, we trained a DeepSequence VAE[34] on homologous Gp2 variants as suggested by Hsu et al.[35] This VAE then inferred one ELBO "evolutionary density score" per Gp2 variant. This ELBO was then concatenated to full and paratope one hot encodings. A 1 × 2 strain identifying OneHot vector was appended to these benchmark representations. These representations were then used to train all top models as listed in Supporting Information Table S2 on our final and key task of protein recombinant yield prediction.

Cross-Validation Protocol. Individual architectures were validated via either Tensorflow[52] or scikit-learn[53] 10 times via k-fold CV; embedding architectures were evaluated using Tensorflow ($k = 3$), whereas top models were evaluated using scikit-learn ($k = 10$). The data within each assessed architecture set (e.g., embedding strategy analysis) was conserved. Hyperopt[59] determined the optimal hyperparameters for each architecture. Validation proceeded across either 50 trials or a maximum of 24 h of computational time; the trial with the lowest predictive error was recorded. The hyperparameters that resulted in optimal performance of an individual architecture pair within a set of embedding model-top model pairs was saved as the embedding model-top model architecture for each architecture pair in that examined set. Tables S1 and S2 summarize the examined architectures across all sets with respective maximum assessed hyperparameter ranges.

Testing Protocol. All architectures within a given embedding model-top model set were retrained with their optimal hyperparameters on the entire CV training set. Held out test data for each architecture set was used to examine each architecture's performance. This independent test set was not used outside of examining architecture test performance.

Metrics. All model training, CV, and testing performances are determined via the MSE of relevant train, CV, and independent test datasets. We additionally define a separate metric of experimental variance: the SSE. This SSE is the sequence-averaged trial-to-trial variance of a single assay or yield (N = 3) measurement. Dividing the experimental variance by N = 3 yields the experimental variance SSE. The experimental variance SSE thus represents the minimum possible observable performance of one of our models on our dataset for our two tasks.

Generation of Hypothesized High Developability Gp2 Variants. A set of 100 paratope sequences—known as walkers—of length 16 were initialized via random selection of all 20 amino acids at every position from a uniform distribution. Gap insertions were further allowed at positions 4–5 and 12–13 to enable encoding of paratopes of total lengths from 12 to 16 amino acids while yielding consistently 16 character sequences. If relevant, the assignment of cysteine to a position was restricted during this initialization process.

These 100 sequences were then used as input to make respective embeddings (e.g., DevRep's 97-dimension paratope embedding) and their predicted developability recorded from the respective best top model for the embedding model.

After initialization, the sampling proceeds by first suggesting a Monte Carlo step and either accepting or rejecting the step according to the respective sampling criteria. In this proposed step, each of k residues in each sequence are mutated with a uniform probability from a pool of allowed amino acids. At the start of sampling, k = 16 (all) positions in the sequence are mutated and changes in stringency (k) are made based on an arbitrary acceptance criterion. This acceptance criterion and general approach vary between our NS[43] and SA[46] schemes, detailed in Supporting Information Section 2.3.

The output of NS is a list of threshold developabilities (and sequences) from which we can compute the DOSs, $g(Y)$, and from it all thermodynamic observables such as the average yield, $\langle Y \rangle_\beta = \sum_Y Y g(Y)\, \mathrm{e}^{-\beta Y} / \sum_Y g(Y)\, \mathrm{e}^{-\beta Y}$, heat capacity $C(\beta) = \beta^2(\langle Y^2 \rangle - \langle Y \rangle^2)$, entropy $S(\beta) = \beta(\langle Y \rangle_\beta - F(\beta))$, and free energy $F(\beta) = -\beta^{-1} \ln(\sum_Y g(Y)\, \mathrm{e}^{-\beta Y})$ where we have taken k_B = 1 everywhere. This information is then used to construct basins of hypothesized highly developable Gp2 variants via disconnectivity graph generation and landscape analysis, described in Supporting Information Section 2.4. In total, 600 Gp2 variants were generatively suggested across our top-performing in-house DevRep and benchmark (OneHot, UniRep) embedding models with a balance between high predicted developability (5/6) vs sequence diversity (1/6).

Experimental Validation of Generated Gp2 Variants. *Gp2 Oligopool.* Oligopools (Twist Bioscience) were designed to transform *E. coli* strains with plasmids encoding our 600 Gp2 variants as previously described.[17]

Note that while our oligopool attempted to produce all 600 Gp2 variants for each of our two strains, only 280/600 and 269/600 variants were characterized for the I^q and SH strains, respectively. We hypothesize this discrepancy in attempted versus successfully measured variants as arising from the stochasticity inherent in sampling from a large combinatorial space of potential oligonucleotides in our pool.

Gp2 Production and Dot Blot Inspection. Dot blot was performed as a modified western blot with higher throughput,[17] detailed in Supporting Information Sections 3.4 and 3.5.

■ ASSOCIATED CONTENT

Supporting Information

The Supporting Information is available free of charge at https://pubs.acs.org/doi/10.1021/acssynbio.3c00196.

> Additional analyses of control and benchmark models, amino acid embeddings, model-suggested high developability protein variants, and supporting methods (PDF)

■ AUTHOR INFORMATION

Corresponding Authors

Benjamin J. Hackel – *Department of Chemical Engineering and Materials Science, University of Minnesota, Minneapolis, Minnesota 55455, United States;* orcid.org/0000-0003-3561-9463; Email: hackel@umn.edu

Stefano Martiniani – *Department of Chemical Engineering and Materials Science, University of Minnesota, Minneapolis, Minnesota 55455, United States; Center for Soft Matter Research, Department of Physics, Simons Center for Computational Physical Chemistry, Departments of Chemistry, and Courant Institute of Mathematical Sciences, New York University, New York, New York 10003, United States;* Email: sm7683@nyu.edu

Authors

Alexander W. Golinski – *Department of Chemical Engineering and Materials Science, University of Minnesota, Minneapolis, Minnesota 55455, United States*

Zachary D. Schmitz – *Department of Chemical Engineering and Materials Science, University of Minnesota, Minneapolis, Minnesota 55455, United States*

Gregory H. Nielsen – *Department of Chemical Engineering and Materials Science, University of Minnesota, Minneapolis, Minnesota 55455, United States*

Bryce Johnson – *Department of Chemical Engineering and Materials Science, University of Minnesota, Minneapolis, Minnesota 55455, United States*

Diya Saha – *Department of Chemical Engineering and Materials Science, University of Minnesota, Minneapolis, Minnesota 55455, United States*

Sandhya Appiah – *Department of Chemical Engineering and Materials Science, University of Minnesota, Minneapolis, Minnesota 55455, United States*

Complete contact information is available at:
https://pubs.acs.org/10.1021/acssynbio.3c00196

Author Contributions

A.W.G. and Z.D.S. are contributed equally to this work. A.W.G., S.M., and B.J.H. designed the research. A.W.G., Z.D.S., B.J., and S.M. performed the computational research. A.W.G. and G.H.N. performed the experimental research. All authors analyzed the data. A.W.G., Z.D.S., G.H.N., B.J.H., and S.M. wrote the paper.

Notes

The authors declare no competing financial interest.

All 45,433 Gp2 variants utilized in this work are described from our previous work[17] provided on Github at https://github.com/HackelLab-UMN/DevRep.

Python scripts and modules used for deep sequencing and model evaluation, as well as datasets to train and evaluate performance are listed above and are available on Github at https://github.com/HackelLab-UMN/DevRep2. Visualization and analysis of models used pandas v0.25.3,[60] Matplotlib v3.3.4,[61] SciPy v1.5.2,[62] seaborn v0.11.1,[63] statsmodels v0.12.1,[64] BioPython v1.74,[65] PyMOL v2.5.0,[66] and NN-SVG.[67]

■ ACKNOWLEDGMENTS

This work was support by the NIH (R01 EB023339 and R01 GM146372) and an NSF Graduate Research Fellowship (to A.W.G.). S.M. also acknowledges support from the Simons Foundation through the Simons Center for Computational Physical Chemistry, as well as partial support from the National Science Foundation (award nos. 2226387 and 2039575). We appreciate support from the University of Minnesota Flow Cytometry Core, University of Minnesota Genomics Center, and the Minnesota Supercomputing Institute at the University of Minnesota.

■ REFERENCES

(1) Gebauer, M.; Skerra, A. Engineered protein scaffolds as next-generation therapeutics. *Annu. Rev. Pharmacol. Toxicol.* **2020**, *60*, 391−415.

(2) Borrebaeck, C. A. K. Precision diagnostics: moving towards protein biomarker signatures of clinical utility in cancer. *Nat. Rev. Cancer* **2017**, *17*, 199−204.

(3) Kennedy, P. J.; Oliveira, C.; Granja, P. L.; Sarmento, B. Antibodies and associates: Partners in targeted drug delivery. *Pharmacol. Ther.* **2017**, *177*, 129−145.

(4) Arbige, M. V.; Shetty, J. K.; Chotani, G. K. Industrial Enzymology: The Next Chapter. *Trends Biotechnol.* **2019**, *37*, 1355−1366.

(5) Engqvist, M. K. M.; Rabe, K. S. Applications of Protein Engineering and Directed Evolution in Plant Research. *Plant Physiol.* **2019**, *179*, 907−917.

(6) Kapoor, S.; Rafiq, A.; Sharma, S. Protein engineering and its applications in food industry. *Crit. Rev. Food Sci. Nutr.* **2017**, *57*, 2321−2329.

(7) Jain, T.; Sun, T.; Durand, S.; Hall, A.; Houston, N. R.; Nett, J. H.; Sharkey, B.; Bobrowicz, B.; Caffry, I.; Yu, Y.; et al. Biophysical properties of the clinical-stage antibody landscape. *Proc. Natl. Acad. Sci.* **2017**, *114*, 944−949.

(8) Raybould, M. I. J. J.; Marks, C.; Krawczyk, K.; Taddese, B.; Nowak, J.; Lewis, A. P.; Bujotzek, A.; Shi, J.; Deane, C. M. Five computational developability guidelines for therapeutic antibody profiling. *Proc. Natl. Acad. Sci.* **2019**, *116*, 4025−4030.

(9) Xu, Y.; Wang, D.; Mason, B.; Rossomando, T.; Li, N.; Liu, D.; Cheung, J. K.; Xu, W.; Raghava, S.; Katiyar, A.; et al. Structure, heterogeneity and developability assessment of therapeutic antibodies. *mAbs* **2019**, *11*, 239−264.

(10) Bailly, M.; Mieczkowski, C.; Juan, V.; Metwally, E.; Tomazela, D.; Baker, J.; Uchida, M.; Kofman, E.; Raoufi, F.; Motlagh, S.; et al. Predicting Antibody Developability Profiles Through Early Stage Discovery Screening. *mAbs* **2020**, *12*, 1743053.

(11) Lobo, S. A.; Bączyk, P.; Wyss, B.; Widmer, J. C.; Jesus, L. P.; Gomes, J.; Batista, A. P.; Hartmann, S.; Wassmann, P. Stability liabilities of biotherapeutic proteins: Early assessment as mitigation strategy. *J. Pharm. Biomed. Anal.* **2021**, *192*, 113650.

(12) Yang, X.; Xu, W.; Dukleska, S.; Benchaar, S.; Mengisen, S.; Antochshuk, V.; Cheung, J.; Mann, L.; Babadjanova, Z.; Rowand, J.; et al. Developability studies before initiation of process development. *mAbs* **2013**, *5*, 787−794.

(13) Romero, P. A.; Arnold, F. H. Exploring protein fitness landscapes by directed evolution. *Nat. Rev. Mol. Cell Biol.* **2009**, *10*, 866−876.

(14) Wolf Pérez, A. M.; Sormanni, P.; Andersen, J. S.; Sakhnini, L. I.; Rodriguez-leon, I.; Bjelke, J. R.; Gajhede, A. J.; De Maria, L.; Otzen, D. E.; Vendruscolo, M.; et al. In vitro and in silico assessment of the developability of a designed monoclonal antibody library. *mAbs* **2019**, *11*, 388−400.

(15) Yang, K. K.; Wu, Z.; Arnold, F. H. Machine-learning-guided directed evolution for protein engineering. *Nat. Methods* **2019**, *16*, 687−694.

(16) Narayanan, H.; Dingfelder, F.; Butté, A.; Lorenzen, N.; Sokolov, M.; Arosio, P. Machine Learning for Biologics: Opportunities for Protein Engineering, Developability, and Formulation. *Trends Pharmacol. Sci.* **2021**, *42*, 151−165.

(17) Golinski, A. W.; Mischler, K. M.; Laxminarayan, S.; Neurock, N.; Fossing, M.; Pichman, H.; Martiniani, S.; Hackel, B. J. High-Throughput Developability Assays Enable Library-Scale Identification of Producible Protein Scaffold Variants. *Proc. Natl. Acad. Sci. U. S. A.* **2021**, *118*, No. e2026658118.

(18) Cerda, P.; Varoquaux, G.; Kégl, B. Similarity encoding for learning with dirty categorical variables. *Mach. Learn.* **2018**, *107*, 1477−1494.

(19) Kawashima, S.; Kanehisa, M. AAindex: amino acid index database. *Nucleic Acids Res.* **2000**, *28*, 374.

(20) Golinski, A. W.; Holec, P. V.; Mischler, K. M.; Hackel, B. J. Biophysical Characterization Platform Informs Protein Scaffold Evolvability. *ACS Comb. Sci.* **2019**, *21*, 323−335.

(21) Alley, E. C.; Khimulya, G.; Biswas, S.; AlQuraishi, M.; Church, G. M. Unified rational protein engineering with sequence-based deep representation learning. *Nat. Methods* **2019**, *16*, 1315−1322.

(22) Rao, R.; Bhattacharya, N.; Thomas, N.; Duan, Y.; Chen, X.; Canny, J.; Abbeel, P.; Song, Y. Evaluating Protein Transfer Learning with TAPE. *Advances in Neural Information Processing Systems*; Curran Associates, Inc., 2019.

(23) Hopf, T. A.; Ingraham, J. B.; Poelwijk, F. J.; Schärfe, C. P. I.; Springer, M.; Sander, C.; Marks, D. S. Mutation effects predicted from sequence co-variation. *Nat. Biotechnol.* **2017**, *35*, 128−135.

(24) Raybould, M. I. J.; Deane, C. M. The Therapeutic Antibody Profiler for Computational Developability Assessment. In *Therapeutic Antibodies: Methods in Molecular Biology*; Houen, G., Ed.; Springer, 2022; pp 115−125.

(25) Kruziki, M. A.; Bhatnagar, S.; Woldring, D. R.; Duong, V. T.; Hackel, B. J. A 45-Amino-Acid Scaffold Mined from the PDB for High-Affinity Ligand Engineering. *Chem. Biol.* **2015**, *22*, 946−956.

(26) Kruziki, M. A.; Sarma, V.; Hackel, B. J. Constrained Combinatorial Libraries of Gp2 Proteins Enhance Discovery of PD-L1 Binders. *ACS Comb. Sci.* **2018**, *20*, 423−435.

(27) Kruziki, M. A.; Case, B. A.; Chan, J. Y.; Zudock, E. J.; Woldring, D. R.; Yee, D.; Hackel, B. J. 64Cu-Labeled Gp2 Domain for PET Imaging of Epidermal Growth Factor Receptor. *Mol. Pharm.* **2016**, *13*, 3747−3755.

(28) Chan, J. Y.; Hackel, B. J.; Yee, D. Targeting Insulin Receptor in Breast Cancer Using Small Engineered Protein Scaffolds. *Mol. Cancer Ther.* **2017**, *16*, 1324−1334.

(29) Bastolla, U.; Porto, M.; Roman, H. E.; Vendruscolo, M. Principal eigenvector of contact matrices and hydrophobicity profiles in proteins. *Proteins* **2005**, *58*, 22−30.

(30) Chou, P. Y.; Fasman, G. D. Secondary structural prediction of proteins from their amino acid sequence. *Trends Biochem. Sci.* **1977**, *2*, 128−131.

(31) Finkelstein, A. V.; Badretdinov, A. Y.; Ptitsyn, O. B. Physical reasons for secondary structure stability: α-Helices in short peptides. *Proteins Struct. Funct. Bioinforma.* **1991**, *10*, 287−299.

(32) McInnes, L.; Healy, J.; Melville, J. UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction. **2020**, arXiv:180203426 Cs Stat.

(33) McInnes, L.; Healy, J.; Astels, S. hdbscan: Hierarchical density based clustering. *J. Open Source Softw.* **2017**, *2*, 205.

(34) Riesselman, A. J.; Ingraham, J. B.; Marks, D. S. Deep generative models of genetic variation capture the effects of mutations. *Nat. Methods* **2018**, *15*, 816−822.

(35) Hsu, C.; Nisonoff, H.; Fannjiang, C.; Listgarten, J. Learning protein fitness models from evolutionary and assay-labeled data. *Nat. Biotechnol.* **2022**, *40*, 1114−1122.

(36) Finn, R. D.; Clements, J.; Eddy, S. R. HMMER web server: interactive sequence similarity searching. *Nucleic Acids Res* **2011**, *39*, W29−W37.

(37) Suzek, B. E.; Wang, Y.; Huang, H.; McGarvey, P. B.; Wu, C. H.; Consortium, U. UniRef clusters: a comprehensive and scalable alternative for improving sequence similarity searches. *Bioinforma. Oxf. Engl.* **2015**, *31*, 926−932.

(38) Devlin, J.; Chang, M.-W.; Lee, K.; Toutanova, K. BERT: Pretraining of Deep Bidirectional Transformers for Language Understanding. **2019**, arXiv:1810.04805.

(39) Finn, R. D.; Bateman, A.; Clements, J.; Coggill, P.; Eberhardt, R. Y.; Eddy, S. R.; Heger, A.; Hetherington, K.; Holm, L.; Mistry, J.; et al. Pfam: the protein families database. *Nucleic Acids Res* **2014**, *42*, D222−D230.

(40) Finn, R. D.; Clements, J.; Arndt, W.; Miller, B. L.; Wheeler, T. J.; Schreiber, F.; Bateman, A.; Eddy, S. R. HMMER web server: 2015 update. *Nucleic Acids Res* **2015**, *43*, W30−W38.

(41) Ma, E. J.; Kummer, A. Reimplementing Unirep in JAX. *bioRxiv* **2020**, DOI: 10.1101/2020.05.11.088344.

(42) Skilling, J. Nested sampling for general Bayesian computation. *Bayesian Anal* **2006**, *1*, 833−859.

(43) Martiniani, S.; Stevenson, J. D.; Wales, D. J.; Frenkel, D. Superposition Enhanced Nested Sampling. *Phys. Rev. X* **2014**, *4*, 031034.

(44) Pártay, L. B.; Bartók, A. P.; Csányi, G. Efficient Sampling of Atomic Configurational Spaces. *J. Phys. Chem. B* **2010**, *114*, 10502−10512.

(45) Burkoff, N. S.; Várnai, C.; Wells, S. A.; Wild, D. L. Exploring the energy landscapes of protein folding simulations with Bayesian computation. *Biophys. J.* **2012**, *102*, 878−886.

(46) Pardalos, P. M.; Mavridou, T. D. Simulated Annealing. In *Encyclopedia of Optimization*; Floudas, C. A., Pardalos, P. M., Eds.; Springer US, 2009; pp 3591−3593.

(47) Lipinski, C. A.; Lombardo, F.; Dominy, B. W.; Feeney, P. J. Experimental and computational approaches to estimate solubility and permeability in drug discovery and development settings. *Adv. Drug Deliv. Rev.* **1997**, *23*, 3−25.

(48) Fowler, D. M.; Fields, S. Deep mutational scanning: a new style of protein science. *Nat. Methods* **2014**, *11*, 801−807.

(49) 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. *Proc. Natl. Acad. Sci.* **2021**, *118*, No. e2104878118.

(50) McConnell, A.; Hackel, B. J. Protein engineering via sequence-performance mapping. *Cell Syst* **2023**, *14*, P656−P666.

(51) cpython/Lib/multiprocessing at 3.11 python/cpython GitHub. https://github.com/python/cpython.

(52) Abadi, M.; Agarwal, A.; Barham, P.; Brevdo, E.; Chen, Z.; Citro, C.; Corrado, G. S.; Davis, A.; Dean, J.; Devin, M.; et al. *TensorFlow, Large-Scale Machine Learning on Heterogeneous Systems*, 2015.

(53) Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V.; et al. Scikit-learn: Machine Learning in Python. *J. Mach. Learn. Res.* **2011**, *12*, 2825−2830.

(54) Harris, C. R.; Millman, K. J.; van der Walt, S. J.; Gommers, R.; Virtanen, P.; Cournapeau, D.; Wieser, E.; Taylor, J.; Berg, S.; Smith, N. J.; et al. Array programming with NumPy. *Nature* **2020**, *585*, 357−362.

(55) Van Rossum, G.; Drake, F. L. *Python 3 Reference Manual (CreateSpace)*, 2009.

(56) Kingma, D. P.; Ba, J. Adam: A Method for Stochastic Optimization. **2017**, arXiv:1412.6980 Cs.

(57) Glorot, X.; Bengio, Y. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics (JMLR Workshop and Conference Proceedings)*; Proceedings of Machine Learning Research, 2010; pp 249−256.

(58) Yeo, I.-K.; Johnson, R. A. A New Family of Power Transformations to Improve Normality or Symmetry. *Biometrika* **2000**, *87*, 954−959.

(59) Bergstra, J.; Komer, B.; Eliasmith, C.; Yamins, D.; Cox, D. D. Hyperopt: a python library for model selection and hyperparameter optimization. *Comput. Sci. Discov.* **2015**, *8*, 014008.

(60) Reback, J.; McKinney, W.; jbrockmendel; Van den Bossche, J.; Augspurger, T.; Cloud, P.; gfyoung; Sinhrks; Klein, A.; Roeschke, M.; et al. *pandas-dev/pandas: Pandas 0.25.3*, 2020.

(61) Hunter, J. D. Matplotlib: A 2D graphics environment. *Comput. Sci. Eng.* **2007**, *9*, 90−95.

(62) Virtanen, P.; Gommers, R.; Oliphant, T. E.; Haberland, M.; Reddy, T.; Cournapeau, D.; Burovski, E.; Peterson, P.; Weckesser, W.; Bright, J.; et al. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nat. Methods* **2020**, *17*, 261−272.

(63) Waskom, M. L. seaborn: statistical data visualization. *J. Open Source Softw.* **2021**, *6*, 3021.

(64) Seabold, S.; Perktold, J. Statsmodels: Econometric and statistical modeling with python. In *9th Python in Science Conference*, 2010.

(65) Cock, P. J. A.; Antao, T.; Chang, J. T.; Chapman, B. A.; Cox, C. J.; Dalke, A.; Friedberg, I.; Hamelryck, T.; Kauff, F.; Wilczynski, B.; et al. Biopython: freely available Python tools for computational molecular biology and bioinformatics. *Bioinformatics* **2009**, *25*, 1422−1423.

(66) Schrödinger, L. *The {PyMOL} Molecular Graphics System*, version~1.8, 2015.

(67) LeNail, A. NN-SVG: Publication-Ready Neural Network Architecture Schematics. *J. Open Source Softw.* **2019**, *4*, 747.