JCIM JOURNAL OF CHEMICAL INFORMATION AND MODELING

pubs.acs.org/jcim Article

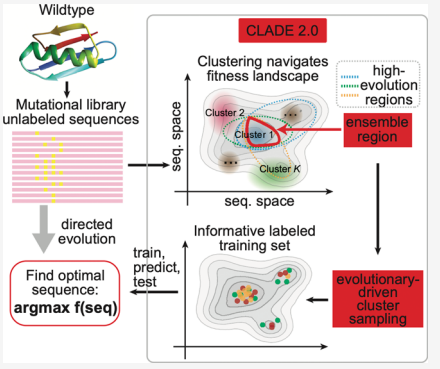
CLADE 2.0: Evolution-Driven Cluster Learning-Assisted Directed Evolution

Yuchi Qiu and Guo-Wei Wei*

Cite This: J. Chem. Inf. Model. 2022, 62, 4629−4641 Read Online

ACCESS | Metrics & More | Article Recommendations | Supporting Information

ABSTRACT: Directed evolution, a revolutionary biotechnology in protein engineering, optimizes protein fitness by searching an astronomical mutational space via expensive experiments. The cluster learning-assisted directed evolution (CLADE) efficiently explores the mutational space via a combination of unsupervised hierarchical clustering and supervised learning. However, the initial-stage sampling in CLADE treats all clusters equally despite many clusters containing a large portion of non-functional mutations. Recent statistical and deep learning tools enable evolutionary density modeling to access protein fitness in an unsupervised manner. In this work, we construct an ensemble of multiple evolutionary scores to guide the initial sampling in CLADE. The resulting evolutionary score-enhanced CLADE, called CLADE 2.0, efficiently selects a training set within a small informative space using the evolution-driven clustering sampling. CLADE 2.0 is validated by using two benchmark libraries both having 160,000 sequences from four-site mutational combinations. Extensive computational experiments and comparisons with existing cutting-edge methods indicate that CLADE 2.0 is a new state-of-art tool for machine learning-assisted directed evolution.



1. INTRODUCTION

Protein functions, broadly referred to as fitness, such as catalytic activity and antibody efficacy, are critical to all living organisms. Protein engineering designs proteins to better serve the needs in real life. Directed evolution (DE), a major approach in protein engineering, optimizes protein fitness by mimicking natural selection via mutagenesis.[1] Mathematically, DE can be formulated as a black-box optimization problem for searching the best sequence x^*

$$x^* = \arg\max_{x \in \mathcal{S}} f(x) \tag{1}$$

Here, $\mathcal{S}$ is the sequence mutational space, and $f(x)$ is an unknown sequence-to-fitness function for sequence x in $\mathcal{S}$. In DE, the mutational space is astronomically large. For example, the combinatorial library consisting of all mutations at expert-selected N mutational sites for a target protein has 20^N sequences.[2,3] To find the global maximal sequence, DE sequentially queries sequences in $\mathcal{S}$ for experimental fitness measurement. However, the experimental measurement is usually expensive and time-consuming. These challenges call for effective searching strategies in DE to navigate the epistatic fitness landscape enriched with local maxima.[2,4,5]

With recent advanced computational tools, especially machine learning models, in silico protein fitness evaluation complements the experimental screening for an expedition.[3,6,7] For example, evolutionary density models provide implicit strategies to predict target protein fitness without specific experimental labels.[8] Particularly, hidden Markov models[9] and Potts models[10] are popular in capturing sequence conservation using multiple sequence alignment (MSA). Local deep learning models such as variational autoencoders (VAEs)[11] offer a similar approach in extracting evolutionary information from MSA. The natural language processing-based protein models such as Transformer[12,13] and long short-term memory[14,15] learn natural selection rules from large sequence databases to predict evolutionary scores of target proteins. Similar to global models, deep MSA Transformer can also predict evolutionary information by training on a large set of MSAs.[16] In addition to unsupervised approaches from the evolutionary models, supervised regression models provide explicit strategies using a set of sequences with experimentally measured fitness to predict the fitness of new sequences. A variety of supervised models have been applied to protein fitness predictions, such as convolutional neural networks,[17] Transformer,[13] and decision tree-based methods.[18,19]

Fueled by the success of computational protein fitness models, machine learning-assisted DE (MLDE) becomes a new strategy in DE for acceleration and systematic exploration.[3,20] MLDE has been widely applied to engineer

Received: August 16, 2022
Published: September 26, 2022

https://doi.org/10.1021/acs.jcim.2c01046
J. Chem. Inf. Model. 2022, 62, 4629−4641

enzyme evolution,[2,21] protein fluorescence,[22] membrane proteins localization,[23] protein thermostability,[24] and antibody efficacy.[25] MLDE is generally an active learning approach that consists of a surrogate model that predicts protein fitness and an acquisition function that determines a query of sequences for the next round of experimental screening.[26] The Gaussian process is an established MLDE method which can balance the exploitation−exploration trade-off.[22−24] Alternatively, other advanced supervised models, which are trained on a randomly selected labeled set to perform greedy search, have shown accurate performance,[2,4] although the informative training set selection is critical to the performance. To avoid the inefficient random sampling in the huge mutational space containing a large portion of non-functional sequences, ftMLDE uses the evolutionary density models to rank sequences and confine the sampling within an informative subspace.[4] Additionally, cluster learning-assisted DE (CLADE) uses unsupervised hierarchical clustering to guide the sampling within more informative subspace with accumulated knowledge of the fitness landscape.[27]

In this work, we proposed an evolution-driven clustering learning-assisted DE, called CLADE 2.0, to improve the inefficient equally sampling in CLADE at the initial stage. We ensemble multiple evolutionary scores to rank sequences to drive robust initial sampling. With no available labeled data, the evolution-driven clustering sampling in CLADE 2.0 targets high-evolution space enriched with informative sequences. At the later stage with labeled data, CLADE 2.0 iteratively refines sampling probabilities and clustering architectures using the labeled data. With a selected informative training set, the final step of CLADE 2.0 executes a greedy search from an ensemble supervised model to pick potentially high-fitness sequences predicted by the model. We benchmark CLADE 2.0 on two benchmark combinatorial libraries with four mutational sites and 160,000 mutations. CLADE 2.0 demonstrates robust and accurate performance compared with other existing advanced MLDE methods in spite of hyperparameter selection.

2. METHODS

2.1. Data Sets. In this work, we use two combinatorial libraries, GB1 and PhoQ, that have almost complete coverage for mutations at four mutational sites. GB1 is a very popular benchmark library, while the PhoQ library has also been used in early MLDE studies.[27,28] PhoQ is considered as an alternative data set. For both data sets, their fitness values were normalized into the range [0, 1] when being applied to CLADE.

The GB1 data set[5] is an empirical fitness landscape for protein G domain B1 (PDB ID: 2GI9) binding to an antibody. Fitness was defined as the enrichment of folded protein bound to the antibody IgG-Fc. This data set contains 149,361 experimentally labeled sequences out of 20^4 = 160,000 possible ones at four amino acid sites (i.e., V39, D40, G41, and V54).

In the PhoQ data set,[29] a high-throughput assay for the signaling of a two-component regulatory system, a PhoQ−PhoP sensor kinase and a response regulator, was developed with a yellow fluorescent protein (YFP) reporter expressed from a PhoP-dependent promoter. Extracellular magnesium concentration stimulates phosphatase or kinase activity of PhoQ, which can be reported by YFP levels. The combinatorial library was constructed at four sites (i.e., A284, V285, S288, and T289) located at the protein−protein interface between the sensor domain and the kinase domain of PhoQ. Two libraries were constructed by using different extracellular magnesium treatments. In each library, the sequences with comparable YFP levels to those of the wild-type were selected by fluorescence-activated cell sorting and used for enrichment ratio calculations. The comparable YFP levels are strictly defined by two thresholds. In this work, we take the enrichment ratios from the library with high extracellular magnesium treatment as fitness. The fitness value correlates to the probability that a variant has fluorescence in the given range, where this range was defined as the wild-type-like activity in the original PhoQ report.[29] The fitness landscape has a nearly complete coverage with 140,517 quality-read sequences out of 20^4 = 160,000. However, the fitness defined in PhoQ is not explicitly correlated to any protein biochemical property. Due to the lack of existing complete combinatorial libraries, we consider PhoQ as an alternative data set for benchmark.

For both GB1 and PhoQ data sets, they are overwhelmed with low- or zero-fitness sequences. By normalizing the fitness to its global maximum, 92% of sequences have fitness lower than 0.01 and 99.3% sequences have fitness lower than 0.3 for GB1. Similarly, there are 92% of sequences having fitness lower than 0.01 and 99.96% of sequences having fitness lower than 0.3 for PhoQ (Supporting Information Figure S1).[27]

2.2. Evolutionary Scores Calculation. The evolutionary density models provide unsupervised approaches to rank fitness from a set of protein mutations. In this work, we calculate five evolutionary scores on combinatorial libraries: profile HMM,[9] ESM-1v Transformer,[12] DeepSequence VAE,[11] EVmutation,[10] and MSA transformer.[16] Except for the global ESM-1v Transformer model, construction of other models requires the MSA of target wild-type protein as input. MSAs are generated by EVcouplings webapp[30] (details see Supporting Information Section S3). When being applied to MSA transformer, the MSAs may need to be subsampled to make the model memory efficient. We used *hhfilter* function in HHsuite package[31] to subsample the alignments by maximizing the diversity as suggested in the original MSA transformer paper.[16] For MSAs of GB1, there are only 56 sequences, and subsampling was omitted. For MSAs of PhoQ, the *−diff* parameter in *hhfilter* was taken as 100, which generates 128 sequences.

The profile HMM is a probabilistic model that captures position-specific information about the amino acid distribution at each site, assuming that the amino acid at a particular position is independent of the amino acid at all other positions.[9] The HMM scores are calculated using profile HMM software. EVmutation model, one type of Potts models, captures site-specific information about amino acids, and it considers the pairwise dependency between amino acids.[10] DeepSequence VAE is a variational autoencoder model that learns sequence distribution from MSA.[11] It uses the evidence lower bound (ELBO) to estimate the sequence log-likelihood to predict the sequence mutational effect. Followed by the original DeepSequence, we trained five VAE models with different random seeds and generated 400 ELBO samples for each model, and the average of all 2000 ELBO samples is used.

ESM-1v[12] is a pretrained Transformer model using global UniRef90 sequence database, while MSA Transformer is another pretrained model using MSAs as training data.[16] Both models can calculate the probability distribution for amino acids at selected positions using the mask-filling protocol.[4,8] The evolutionary scores are calculated as the

https://doi.org/10.1021/acs.jcim.2c01046
J. Chem. Inf. Model. 2022, 62, 4629−4641

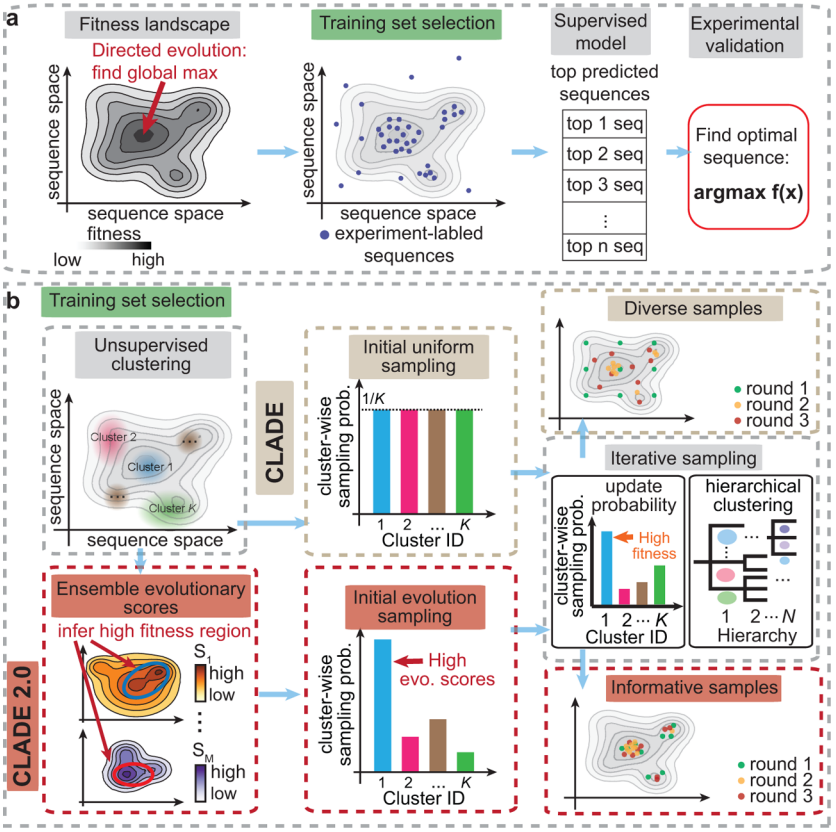


Figure 1. Schematic graph. (a) MLDE searches for a global max sequence in the fitness landscape. It consists of two major steps: training set selection and supervised greedy search to select top sequences. (b) CLADE and CLADE 2.0 are two MLDE strategies that are able to select an informative training set. Initial unsupervised clustering is performed to divide the mutational space into several clusters. CLADE initially uses uniform sampling over clusters. The selected labeled sequences are used to update sampling probabilities and new hierarchical clustering, and the training set is iteratively selected. Evolutionary scores predict fitness landscape in an unsupervised approach and initiate the sampling probability for CLADE 2.0. The later stage clustering sampling of CLADE 2.0 uses the same protocol of CLADE.

pseudo-log-likelihoods by assuming that the distribution of each residue is independent. Specifically, for a given sequence $s = s_1, s_2, ...s_L$, the log-likelihoods at i position is given as log $P(m_i|s_{\text{const}})$, where s_{const} is the sequence s excluding the masked position m_i. In the combinatorial library, the log-likelihoods at multiple positions, $m_{i_1}, m_{i_2},...m_{i_k}$, are estimated by the sum of the log-likelihood of each single mutation, which is the pseudo-log likelihoods

$$\log\ P(m_{i_1}, m_{i_2}, ..., m_{i_k}|s_{\text{const}}) \approx \sum_{j=1}^{k} \log\ P(m_{i_j}|s_{\text{const}}) \quad (2)$$

For ESM-1v, there are five available models trained on five random seeds. The pseudo-log likelihoods are evaluated on all models and averaged.

2.3. Machine Learning-Assisted Directed Evolution. MLDE is a general two-step framework to exploit protein fitness[2,4] for DE (Figure 1a). First, it queries a set of sequences in the mutational space $\mathcal{S}$ for experimental fitness measurement. The set of labeled sequences is taken as the training data for the downstream supervised prediction. CLADE and CLADE 2.0 presented in this work both belong to MLDE but use different training set selection strategies. In this work, physicochemical features are used to encode the sequences in the mutational space[27] (Supporting Information Section S2). The supervised model learns from the training set and predicts fitness for all sequences in the mutational space. At the last step of MLDE, the sequences with top predicted fitness from the supervised model are experimentally screened to exploit the fitness. Random sampling is a naive approach in generating training set which showed significant improvement over traditional DE.[4] The ftMLDE method uses a zero-shot strategy to constrain the random sampling within an informative subspace, and it substantially improves the performance.[4]

For the supervised model, an ensemble model by integrating predictions from multiple regression models was used to accommodate the various sizes of training set.[4] In this work, we construct an ensemble of 17 regression models optimized by Bayesian hyperparameter optimizations.[32] The 17 regression models include scikit-learn models,[33] Keras neural network models,[34] and XGBoost models.[35] The five-fold cross validation is performed on training data and used to evaluate the performance of each model measured by mean square errors. Bayesian hyperparameter optimizations are performed to find the best-performing hyperparameters for each model. After hyperparameter optimizations, the top three models are selected and averaged to predict the fitness of unlabeled sequences. Details are given in the Supporting Information Section S4 and Tables S2 and S3. The top M sequences predicted by MLDE are experimentally screened. Then, the performance of our model is evaluated on the union of training set and the top M sequences.

2.4. Cluster Learning-Assisted Directed Evolution. To improve MLDE performance, CLADE generates a more

https://doi.org/10.1021/acs.jcim.2c01046
J. Chem. Inf. Model. 2022, 62, 4629−4641

informative training set using a clustering sampling via unsupervised hierarchical clustering[27] (Figure 1b). Particularly, *K*-means[36] is used here.

Sequences are first encoded by physicochemical features[37−40] (Supporting Information Section S2). An unsupervised clustering, *K*-means, is performed to divide the mutational space into several subspaces. To select a sequence for experimental screening, one cluster is selected according to the predefined cluster-wise sampling probabilities and random sampling is performed within the selected cluster. The clustering sampling explores clusters enriched with high-fitness sequences. The average fitness for each cluster can be estimated from the labeled samples selected from previous stages. The cluster-wise sampling probabilities are set to be proportional to the estimated average fitness over clusters (Figure 1b). Specifically, in k-th cluster at h-th hierarchy, the sampling probability is given by

$$P_k^{(i)} = \frac{\frac{1}{\#\tilde{C}_k^{(h)}}\sum_{j\in\tilde{C}_k^{(h)}} y_j}{\frac{1}{\sum_l \#\tilde{C}_l^{(h)}}\sum_l \sum_{j\in\tilde{C}_l^{(h)}} y_j} \tag{3}$$

where I is the index set of selected sequences, and $\tilde{C}_l^{(h)}) \subset I$ is the index set of selected sequences l-th cluster at h-th hierarchy. y_j is the fitness of j-th sequence.

With maximum hierarchy N, increment of clusters at h-th ($h \leq N$) hierarchy is given by K_h. The total number of clusters at maximum hierarchy is the sum of these numbers $\sum_{h=1}^{N} K_h$. To further explore high-fitness clusters, hierarchical clustering divides clusters into subclusters, and the increments of new subclusters for parent clusters are proportional to their estimated average fitness (Figure 1b). The k-th parent cluster at $(h-1)$-th hierarchy will be divided into $L_k^{(h)}$ subclusters at h-th hierarchy, and $L_k^{(i)}$ is given by

$$L_k^{(h)} = \begin{cases} [P_k^{(h)} K_i] + 1, & \text{if } k \neq k_0 \\ K_h - \sum_{j\neq k_0} [P_j^{(h)} K_h] + 1, & \text{if } k = k_0 \end{cases} \tag{4}$$

where $k_0 = \arg\max_k \frac{1}{\#\tilde{C}_k^{(h)}}\sum_{j\in C_k^{(h)}} y_j$ is the index of the cluster having the largest average fitness from selected sequences over all clusters. Here, $[x]$ represents the largest integer not greater than x.

By introducing the key components mentioned above, we summarize the flow of CLADE. The clustering sampling has $N+1$ hyperparameters, including maximum hierarchy N and the increment of clusters at each hierarchy K_h. The batch size, $\mathrm{NUM}_{\mathrm{batch}}$, is taken to be the number of sequences being screened in parallel during the experiment. The batch size decides the frequency for updating sampling probability and clusters at new hierarchy, and a lower batch size usually leads to more accurate CLADE prediction but higher cost in the experiment. A typical batch size is 96 for a medium throughput, which is also used in this work, followed by the small 96-well plate commonly seen in many experimental systems.[2,22] At the first round of selection, the first-round clustering is performed to divide the space into K_1 clusters. $\mathrm{NUM}_{1\mathrm{st}}$ sequences are randomly picked over clusters to have a rough coverage of all clusters. Cluster-wise sampling probability is updated every batch according to eq 3. A new hierarchy of clusters is generated after every $\mathrm{NUM}_{\mathrm{hierarchy}}$ sequence is screened until reaching the maximum hierarchy N. In particular, we do not perform the second hierarchical clustering after $\mathrm{NUM}_{1\mathrm{st}}$ sequences are collected but after $\mathrm{NUM}_{1\mathrm{st}} + \mathrm{NUM}_{\mathrm{hierarchy}}$. This allows the sampling to capture the cluster-wise average fitness more accurately for the large space. The labeled data generated from clustering sampling is then taken as the training data for the downstream ensemble supervised model, which is the one used in MLDE. Top M predicted samples are screened experimentally. These numbers, $\mathrm{NUM}_{1\mathrm{st}}$, $\mathrm{NUM}_{\mathrm{hierarchy}}$, $\mathrm{NUM}_{\mathrm{train}}$, and M, are all required to be multiples of batch size $\mathrm{NUM}_{\mathrm{batch}}$. In this work, they are all taken as 96, except for $\mathrm{NUM}_{\mathrm{train}}$, which is taken as 384.

2.5. CLADE 2.0: Ensemble Evolutionary Score Enhances Initial Sampling in CLADE. At the initial sampling stage, no labeled samples are available to estimate the fitness heterogeneity over space. CLADE initially takes the uniform cluster-wise sampling probabilities. As a result, early-stage sampling inefficiently explores the large non-functional space enriched with low- and zero-fitness sequences. Alternatively, ftMLDE proposed a useful zero-shot strategy that employs an evolutionary score to rank the sequences in the mutational space,[4] and the sampling is performed in a small subspace consisting of top L sequences.

In this work, we proposed CLADE 2.0 that uses an ensemble evolutionary score to initiate early-stage sampling (Figure 1b). Five evolutionary scores are used to build the ensemble score: profile HMM,[9] ESM-1v Transformer,[12] DeepSequence VAE,[11] EVmutation,[10] and MSA transformer.[16] The evolutionary scores have been shown to have a high correlation with protein fitness.[4,8] With no available labeled data, the evolution-driven clustering sampling initiates cluster-wise sampling probabilities when a set of clusters is given. Suppose there are K clusters, the i-th evolutionary score alone can first define its cluster-wise sampling probability, $P_k^{(i)}$, as the softmax function of the cluster-wise average scores, $S_k^{(i)}$, at k-th cluster

$$S_k^{(i)} = \frac{1}{\#C_k}\sum_{j\in C_k} s_j^{(i)}$$

$$P_k^{(i)} = \frac{e^{\alpha S_k^{(i)}}}{\sum_j^K e^{\alpha S_j^{(i)}}} \tag{5}$$

where C_k is the index set of all sequences in k-th cluster. The evolutionary score, $s_j^{(i)}$, is linearly normalized into range $[0, 1]$ for all sequences in the mutational space. Here, α is the hyperparameter in softmax function where a larger value leads to higher sampling probabilities on clusters with a high average score. In this work, we take $\alpha = 10$. The ensemble of five evolutionary scores takes a weighted sum of the individual cluster-wise sampling probabilities for all scores. The weight is defined as the heterogeneity index. First, for the i-th evolutionary score, its cluster-wise average scores, $S_k^{(i)}$, are sorted in ascending order with permutation map τ. Then, linear regression is used to fit the average cluster-wise scores in ascending order, and the slope of the linear model is taken as the heterogeneity index

$$H^{(i)} = \operatorname*{argmin}_{\omega} \sum_{j=1}^{K} (S_{\tau(i)} - \omega x_j)^2 \tag{6}$$

https://doi.org/10.1021/acs.jcim.2c01046
J. Chem. Inf. Model. 2022, 62, 4629−4641

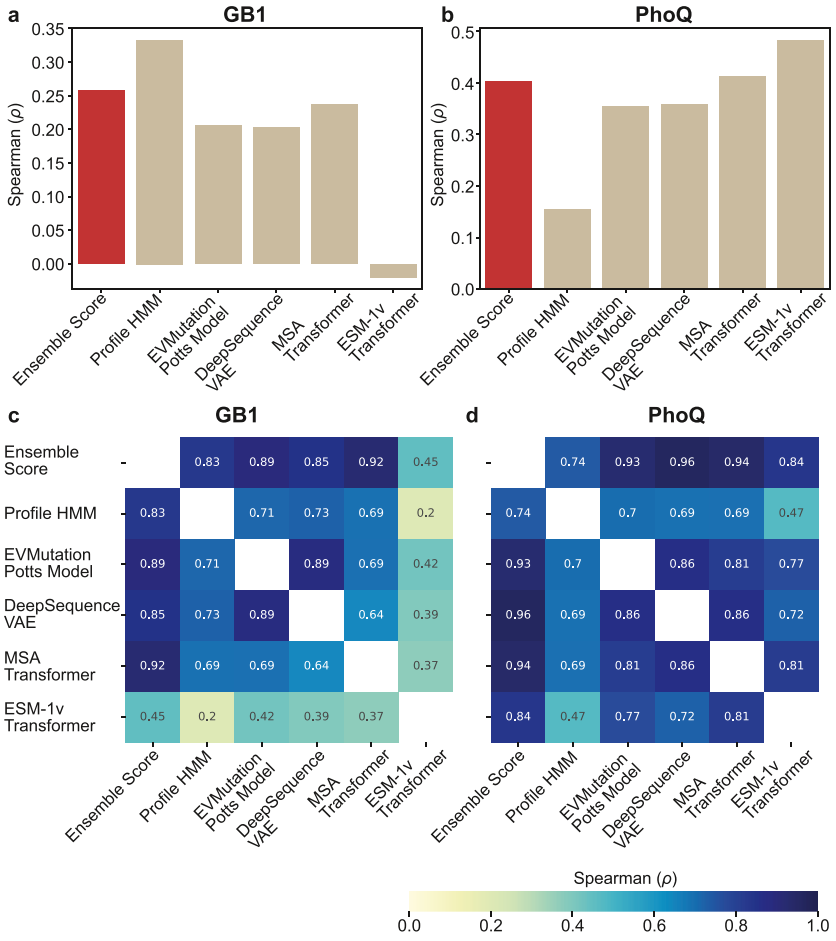


Figure 2. Ensemble evolutionary scores accurately rank fitness landscape. The ensemble score is obtained from the weighted sum of all five evolutionary scores using $K_{ev} = 4$ in calculating the heterogeneity index. Spearman's correlation ρ between fitness and an evolutionary score for (a) GB1 data set and (b) PhoQ data set. The pairwise Spearman's correlation ρ between different evolutionary scores for (c) GB1 data set and (d) PhoQ data set.

where $x_j = (j - 1)/(K - 1)$ uniformly distributes in the unit interval $[0, 1]$. The heterogeneity index, $H^{(i)}$, is between 0 and 1, and a higher value indicates that the corresponding score is more important to the clustering and the recognition of fitness heterogeneity. Taken together, the ensemble cluster-wise sampling probability for k-th cluster is given as the normalized weighted sum of probabilities of all five evolutionary scores

$$P_k = \frac{\sum_{i=1}^{5} H^{(i)} P_k^{(i)}}{\sum_{k=1}^{K} \sum_{i=1}^{5} H^{(i)} P_k^{(i)}} \tag{7}$$

CLADE 2.0 has $N + 2$ hyperparameters. It includes $N + 1$ hyperparameters from CLADE, which are the maximum hierarchy N and the increment of clusters at each hierarchy K_h. Additionally, CLADE 2.0 has an extra hyperparameter K_{ev} that is the number of clusters at the initial 0-th hierarchy given by evolutionary scores. At the initial sampling stage without labeled data, CLADE 2.0 uses evolutionary scores to encode the sequences and divides the space into K_{ev} clusters. The initial evolution-driven cluster-wise probabilities are calculated using eq 7. To focus on the high-evolution clusters, the hierarchical clustering with cluster increment K_1 is built on K_{ev} clusters. Indeed, the physicochemical feature is used for clustering at higher hierarchy when labeled data are available to present its partial supervised manner. The first hierarchy calculates the numbers of subclusters for parent clusters in eq 4 using the initial evolution-driven cluster-wise probabilities. The initial sampling uses $K_{ev} + K_1$ clusters with the cluster-wise sampling probabilities given in eq 7. After the first NUM_{1st} sequences are screened, the procedure of CLADE 2.0 is the same with CLADE for updating sampling probabilities and constructing new clusters at new hierarchy. Especially, CLADE 2.0 also uses the physicochemical feature for clustering since the high-dimension feature may have better fitting ability and the later stage sampling has a supervised manner by utilizing labeled data.

2.6. Evaluation Metrics. MLDE, CLADE, and CLADE 2.0 are all evaluated on three sets, including training set, the top M predicted sequences, and their union. In selecting top M predicted sequences, only sequences that could be constructed by recombination of sequences in the training set are considered. This enhances the confidence of predictions by reducing the extrapolations, especially when a less diverse training set is available. We mainly assess three metrics: mean fitness, max fitness, and global maximal fitness hit rate. The mean fitness is the average fitness for sequences in the given set. The max fitness is the maximal fitness found in the given set. The global maximal fitness hit rate calculates the frequency that the global maximal sequence is successfully picked by the method, which is counted in multiple independent repeats. In this work, we take 200 repeats.

https://doi.org/10.1021/acs.jcim.2c01046
J. Chem. Inf. Model. 2022, 62, 4629−4641

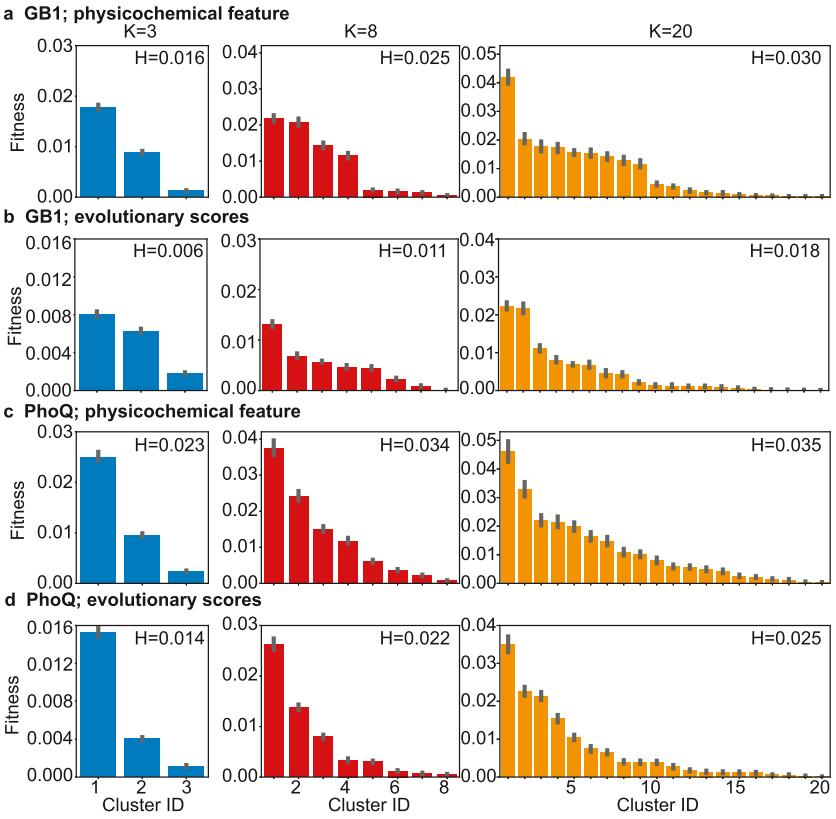


Figure 3. Unsupervised *K*-means clustering captures fitness heterogeneity. *K*-means clustering divides the space into *K* clusters. Plots show average fitness over clusters, where clusters are listed in the descending order of their fitness. Heterogeneity index (H) is shown in each subplot. (a) GB1 data set using the physicochemical feature. (b) GB1 data set using five evolutionary scores as features. (c) PhoQ data set using physicochemical features. (d) PhoQ data set using five evolutionary scores as features.

3. RESULTS

3.1. Ensemble Score is Robust and Accurate in Ranking Fitness. First, we assess individual evolutionary scores in the ranking fitness landscape. We use Spearman's correlation (ρ) to quantify the rank correlation. For the GB1 data set, four local evolutionary scores have high correlation above 0.2 in Spearman's correlation (Figure 2a). Especially, profile HMM performs best among them with $\rho = 0.33$. However, the ESM-1v score has a low and negative $\rho = -0.02$. For the PhoQ data set, all five evolutionary scores have high and positive correlations (Figure 2b). In contrast to GB1, profile HMM has the lowest $\rho = 0.15$ and ESM-1v score has the highest $\rho = 0.48$. Interestingly, the best individual method in one data set performs the worst in another data set. Therefore, the performance of the evolutionary score is sensitive to data sets. To extend the generalization of the evolutionary score, we propose the ensemble score by taking a weighted sum of all scores using the heterogeneity index, as shown in eq 6. The ensemble score integrates all evolutionary scores, and the less informative score can be identified by the heterogeneity index and contributes less to the ensemble one. As a result, the ensemble score achieves high ρ with $\rho = 0.26$ and $\rho = 0.40$ for GB1 and PhoQ, respectively, while the best individual score on one data set has a poor performance on another data set. Although the ensemble score ranks the second and third best on GB1 and PhoQ, it never underperforms one score on both data sets.

The heterogeneity index can automatically assign lower weights to a less informative score. As a result, the ensemble score using the heterogeneity index as weights has a low correlation with the poor performing individual scores. For example, it has the lowest $\rho = 0.45$ with ESM-1v Transformer, while ρ is above 0.83 for other scores on GB1 (Figure 2c). Similarly, it has the lowest $\rho = 0.74$ with profile HMM, while ρ is above 0.84 for other scores on PhoQ (Figure 2d). Moreover, the ensemble score inherits key information from all individual scores. For any evolutionary score, its correlation ρ always achieves highest values with the ensemble score than other individual scores. Indeed, our proposed construction of the ensemble score can selectively integrate advanced scores to perform robust predictions on protein fitness.

3.2. Evolutionary Scores Capture Fitness Heterogeneity via Unsupervised Clustering. The fitness landscape is highly heterogeneous with large portions of zero- and low-fitness sequences (Supporting Information Figure S1). The unsupervised clustering can capture such fitness heterogeneous. Here, we examine the heterogeneity levels revealed by different featurizations (Figure 3).

Physicochemical features were used to encode sequences in combinatorial libraries for CLADE. *K*-means clustering was applied to reveal the fitness heterogeneity over clusters where average fitness in clusters has a non-uniform distribution[27] (Figure 3a,c). In CLADE 2.0, evolutionary scores are first used to encode the sequences at the initial stage. The initial sampling performs the *K*-means clustering using the evolutionary scores. The fitness heterogeneity can also be revealed for both GB1 and PhoQ data sets (Figure 3b,d). The level of fitness heterogeneity over clusters can be quantified by the heterogeneity index (H) in eq 6 using the cluster-wise average

https://doi.org/10.1021/acs.jcim.2c01046
J. Chem. Inf. Model. 2022, 62, 4629−4641

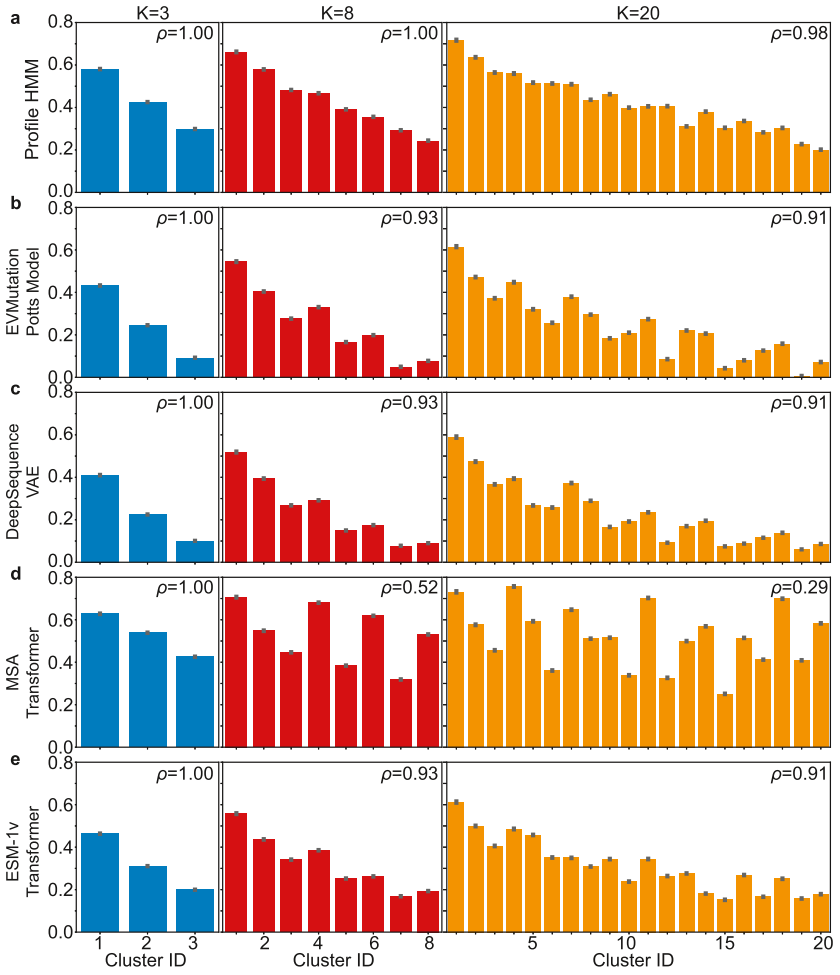


Figure 4. Evolutionary scores capture fitness heterogeneity revealed by *K*-means for GB1 data set. *K*-means is performed using five evolutionary scores as features. Clusters are listed in the descending order of their average fitness, as shown in Figure 3. The *y*-axis shows the average evolutionary score for (a) Profile HMM, (b) EVmutation, (c) DeepSequence VAE, (d) MSA Transformer, and (e) ESM-1v Transformer. Spearman's correlation between the cluster-wise average fitness and the cluster-wise average evolutionary score is shown in each graph.

fitness. The evolutionary score encoding achieves a much higher heterogeneity index than physicochemical features do (Figure 3). This result indicates that evolutionary scores can better divide the mutational space into distinguished fitness levels. In addition, the larger number of cluster K may be more efficient to classify the fitness landscape as its heterogeneity index increases when K increases.

A single evolutionary score alone is able to infer fitness (Figure 2). However, ρ is not high enough to have a reliable prediction on a single sequence. Indeed, we rank fitness at a low resolution for average fitness over clusters. By clustering the space into several subspaces using evolutionary features, we examine the Spearman's correlation between average fitness and average evolutionary score over clusters. By listing the clusters in an descending order for their average fitness, the distribution of the average evolutionary score over clusters can also provide a visualization of the rank correlation (Figures 4 and 5). With a small number of clusters (e.g., $K = 3$), every evolutionary score achieves the perfect correlation (i.e., $\rho = 1$) for both GB1 and PhoQ data sets. As the number of clusters increases, the rank correlation goes down. In particular, MSA Transformer fails to recognize the descending average fitness with a noisy distribution of average scores on GB1 (Figure 4d). A similar poor performance from profile HMM was found on PhoQ (Figure 5a). In the initial sampling in CLADE 2.0, the evolutionary score can rank fitness accurately at the low resolution only with small number of clusters.

3.3. Evolution-Driven Clustering Sampling Captures the Cluster-wise Fitness Heterogeneity. After we showed evolutionary scores accurately rank fitness in an unsupervised manner, we present the simulation of the evolution-driven clustering sampling to generate a training set for CLADE 2.0 (Figure 6). In this simulation, we use GB1 data set as an example. We pick the initial number of clusters $K_{ev} = 4$ and $N = 3$ hierarchy with the same increment of the number of clusters at each hierarchy $K_1 = K_2 = K_3 = 4$. The initial sampling selects $NUM_{1st} = 96$ sequences. The batch size and the hierarchical batch are all taken as 96: $NUM_{batch} = NUM_{hierarchy} = 96$. The number of training data is $NUM_{train} = 384$.

The initial 0-th hierarchical evolution-driven clustering divides the space into $K_{ev} = 4$ clusters using evolutionary scores. Among these four clusters, three contain the majority of sequences with low average fitness (cluster 14−16). The evolution-driven cluster-wise sampling probabilities in eq 7 successfully identify this high-fitness cluster, and the first hierarchical clustering only divides the high-fitness cluster into $K_1 + 1 = 5$ subclusters. The initial sampling is performed on

https://doi.org/10.1021/acs.jcim.2c01046
J. Chem. Inf. Model. 2022, 62, 4629−4641

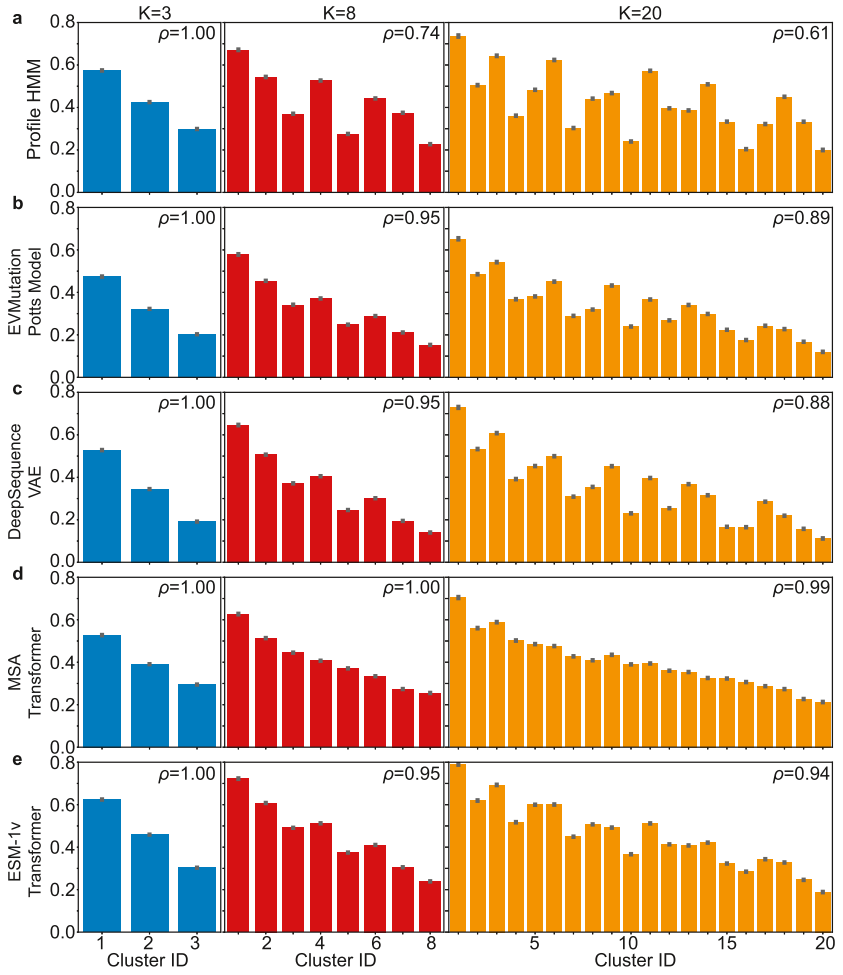


Figure 5. Evolutionary scores capture fitness heterogeneity revealed by *K*-means for PhoQ data set. *K*-means is performed using five evolutionary scores as features. Clusters are listed in the descending order of their average fitness, as shown in Figure 3. The *y*-axis shows the average evolutionary score for (a) Profile HMM, (b) EVmutation, (c) DeepSequence VAE, (d) MSA Transformer, and (e) ESM-1v Transformer. Spearman's correlation between the cluster-wise average fitness and the cluster-wise average evolutionary score is shown in each graph.

these $K_{ev} + K_1 = 8$ clusters to select 96 sequences. This initial sequence selection oversamples 89.6% = 86/96 sequences within the $K_1 + 1 = 5$ subclusters generated at the first hierarchy that only include a minority of sequences in the mutational space (i.e., 9.4% = 14,044/149,361). The cluster with lowest average fitness (i.e., cluster 16) containing 29.9% = 44,712/149,361 sequences is never sampled in the initial sampling and is excluded from the entire sampling process. The initial evolution-driven sampling shows that evolutionary scores can accurately identify high-fitness clusters without using labeled data.

After the initial sampling, CLADE 2.0 follows the same clustering sampling in CLADE using the previously selected sequences to update sampling probabilities and new clustering hierarchy. Specifically, the clustering sampling tends to select sequences in clusters with high average fitness from the previously selected sequences. The hierarchical clustering also tends to divide the high-fitness clusters. As a result, at the maximal third hierarchy, the newly generated five clusters (i.e., clusters 1−5) are ranked among top six high-fitness clusters (Figure 6d). By accurately identifying the high-fitness clusters, the clustering sampling largely oversamples the high-fitness clusters. The five clusters newly generated at the maximum hierarchy $N = 3$ (i.e., clusters 1−5) selects 28% = 109/384 sequences of the training set, while these clusters contain an extremely small portion sequences of the mutational space (0.64% = 960/149,361). As a result, the evolution-driven clustering sampling in CLADE 2.0 efficiently selects informative training set in a limited size of subspace containing high-fitness sequences. The heterogeneity of cluster-wise average fitness can be recapitulated by the selected sequences.

3.4. CLADE 2.0 Exhibits Accurate and Robust Performance. Here we perform full CLADE 2.0 simulations by combining evolution-driven clustering sampling and the downstream supervised learning. We compare CLADE 2.0 with other methods in optimizing fitness. Since many methods have multiple hyperparameters, we explore them extensively. In the experimental application, only one set of hyperparmeters can be used. Indeed, in the comparisons, we not only look at the best performing hyperparameters for each method, but also focus on the worst performing hyperparameters to evaluate the method's robustness.

First we compare CLADE 2.0 with CLADE. For both methods, we set equal increments of clusters $K_1 = K_2 = K_3$ and explore five values with 10, 20, ..., 50. For CLADE 2.0, we only explore small values of K_{ev} below 10, since large K_{ev} hinders the ability of evolutionary scores in capturing the cluster-wise fitness heterogeneity (Figures 5 and 6). For the GB1 data set,

https://doi.org/10.1021/acs.jcim.2c01046
J. Chem. Inf. Model. 2022, 62, 4629−4641

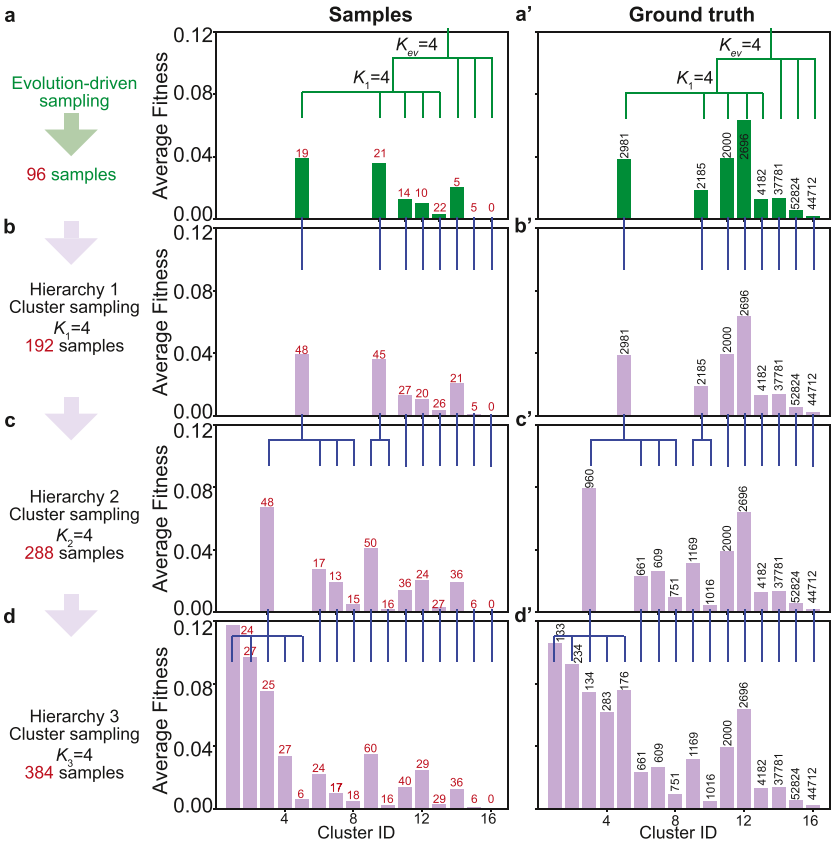


Figure 6. Simulation of procedure of evolution-driven clustering sampling in CLADE 2.0. The simulation is performed with K_{ev} = 4 and K_1 = K_2 = K_3 = 4. Cluster-wise average fitness for (a−d) selected samples and (a′−d′). Number above each bar shows the number of samples selected in the cluster (red font) or number of samples contained in the cluster (black font).

once the evolution-driven sampling is introduced, CLADE 2.0 shows a clear improvement over CLADE for the max fitness and the global maximal fitness hit rate (Figure 7a,b). With K_{ev} between 4 and 8, the performance of CLADE 2.0 shows relatively better performance. For the PhoQ data set, CLADE 2.0 consistently shows improvement over CLADE (Figure 7c,d) with K_{ev} = 2, 3. When K_{ev} increases, the performance tends to go down. Especially, CLADE 2.0 underperforms CLADE for K_{ev} = 6.

Next, we compare CLADE 2.0 with other existing machine learning methods for DE with their most optimistic and pessimistic performance among the hyperparameters explored according to their expected max fitness (Table 1). First, we perform comparison with MLDE using random sampling for training set selection, named random sampling-based MLDE, which also serves as the standard baseline. Furthermore, we include comparisons with ftMLDE that uses single evolutionary score to restrict the random sampling within a subspace containing top sequences given by hyperparameter "Sampling threshold", where EVmutation and MSA Transformer scores are tested. Finally, we include a comparison with CLADE. For optimistic performance, CLADE 2.0 exhibits dominant performance for both GB1 and PhoQ with global maximal fitness hit rate 88 and 27.5%, respectively. Its global maximal fitness hit rate shows 4.7- and 3.6-fold increase over the random sampling-based MLDE. With the assistance of evolutionary score, ftMLDE improves the random sampling-based MLDE but the optimistic results underperform CLADE 2.0. We further look into the pessimistic performance. On the GB1 data set, the worst performance for CLADE 2.0 can still achieve 58% global maximal fitness hit rate, which remarkably outperforms MLDE and the optimistic CLADE. Performance of ftMLDE varies depending on the evolutionary score used, and both pessimistic ftMLDEs underperform CLADE 2.0. On the PhoQ data set, the pessimistic CLADE 2.0 suffers reductions on both global maximal fitness hit rate and expected max fitness from its optimistic results. However, interestingly, the pessimistic and optimistic CLADE 2.0 have similar levels of expected mean fitness (Table 1). In contrast to the highest expected mean fitness achieved by CLADE 2.0 among pessimistic models, pessimistic CLADE 2.0 even underperforms random sampling-based MLDE and pessimistic CLADE on global maximal fitness hit rate and expected max fitness. The performance of pessimistic ftMLDE highly depends on the evolutionary score used: using EVmutation, it achieves the highest expected max fitness and global maximal fitness hit rate among pessimistic methods, but using MSA Transformer, it results in the worst performance with 0% global maximal fitness hit rate.

Our CLADE 2.0 shows state-of-art performance on the GB1 data set despite the optimistic or pessimistic results, highlighting its accurate and robust performance. The GB1 data set serves as the primary benchmark to showcase powerful results. The PhoQ is an alternative data set, where taking the fitness measured by enrichment ratio does not indicate any protein functions. Indeed, the evolutionary score cannot accurately capture the enrichment ratio, leading to the poor pessimistic CLADE 2.0 performance. Such fitness in PhoQ usually will not be used for practical protein engineering. Indeed, CLADE 2.0 with the best optimistic performance on all metrics and best

https://doi.org/10.1021/acs.jcim.2c01046
J. Chem. Inf. Model. 2022, 62, 4629−4641

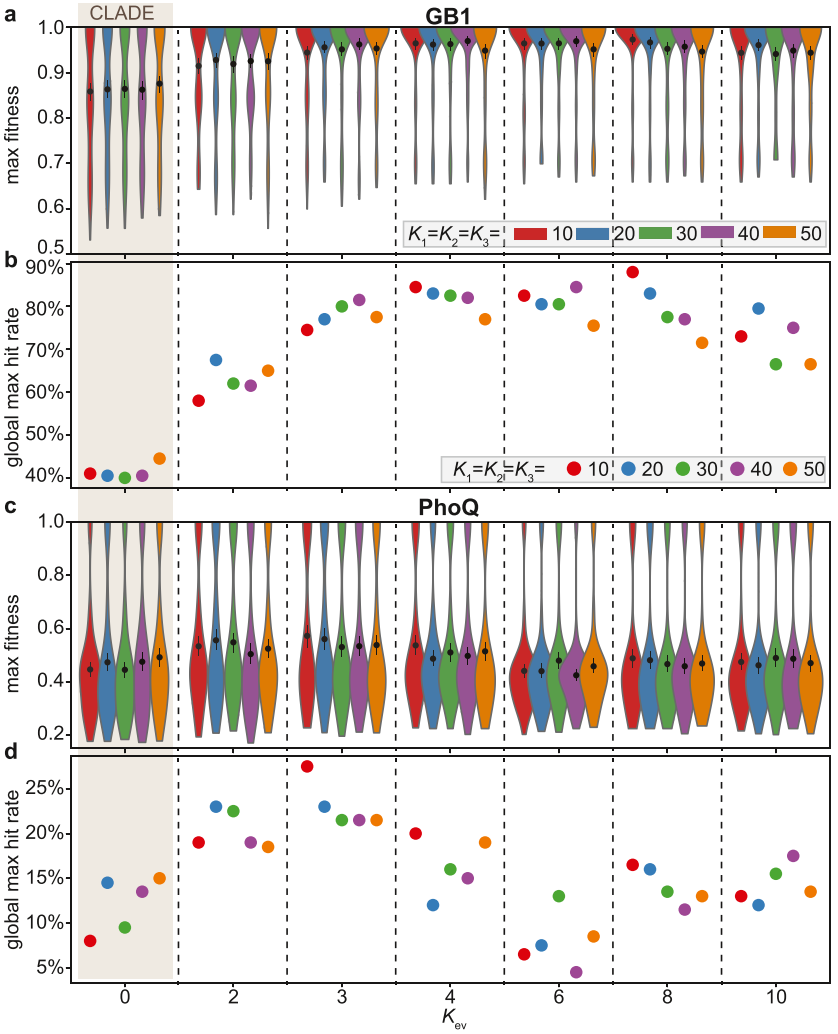


Figure 7. CLADE 2.0 performance on the GB1 data set for (a) expected max fitness and (b) global maximal fitness hit rate; and on the PhoQ data set for (c) expected max fitness and (d) global maximal fitness hit rate. The case with K_{ev} = 0 is equivalent to CLADE. In (a) and (c), violin plots show the kernel density of the max fitness obtained from 200 independent CLADE 2.0 repeats, and the dots show the average with 95% confidence interval shown in the error bars.

pessimistic performance on expected mean fitness confirms its utility and robustness for DE.

4. CONCLUDING REMARKS

MLDE is a powerful approach for protein engineering. One of the most effective MLDE tools is CLADE, which iteratively optimizes protein fitness by navigating a large combinatorial library. In this study, we introduce CLADE 2.0 that takes the advantage of evolutionary scoring to further enhance CLADE. We first show multiple evolutionary scores that can accurately rank protein fitness in an unsupervised manner. Then, we develop an ensemble of five evolutionary scores to capture fitness heterogeneity revealed by unsupervised clustering. The ensemble evolutionary score is designed to carry out the evolution-driven clustering sampling. At the initial stage, sequences are selected within the high-evolution space for experimental screening. Further sampling stages update sampling and clustering using the collected labeled data. The last step invokes an ensemble supervised learning model to exploit fitness via a greedy search. Two benchmark libraries, GB1 and PhoQ, are employed to validate the proposed CLADE 2.0 for ranking global maximal sequences. Performance is compared with that of many cutting-edge methods in MLDE, indicating that CLADE 2.0 is a new state-of-art method for MLDE.

We further summarize the difference between CLADE and CLADE 2.0. The major difference relies on the initial stage in performing clustering and sampling. CLADE uses physicochemical features for clustering, and the initial sampling equally selects mutations over clusters. CLADE 2.0 performs clustering with two hierarchies at the initial stage where the first hierarchy uses evolutionary scores as features and the second hierarchy uses physiochemical features to further partition the space. The initial sampling in CLADE 2.0 no longer equally samples clusters. It oversamples the clusters with a high ensemble evolutionary score. The clusters with low evolutionary scores are rarely selected and even excluded in the exploration. At the later stages, CLADE and CLADE 2.0 share the same strategy in building clusters and selecting mutations. However, the later-stage selection highly relies on the samples and clustering architectures from the initial stage. Indeed, the training data selected by CLADE and CLADE 2.0 have distinct sequence identity (Supporting Information Figures S2 and S3).

https://doi.org/10.1021/acs.jcim.2c01046
J. Chem. Inf. Model. 2022, 62, 4629−4641

Table 1. Comparisons with CLADE 2.0[a]

data set	method	expected max fitness	expected mean fitness	global maximal fitness hit rate	notes
GB1	MLDE (random sampling)	0.774	0.305	18.6%	
			Optimistic Performance		
	CLADE	0.876	0.418	44.5%	$K_1 = K_2 = K_3 = 50$
	ftMLDE (EVmutation)	0.935	0.418	73.0%	sampling threshold = 12,800
	ftMLDE (MSA Transformer)	0.943	0.422	74.5%	sampling threshold = 16,000
	CLADE 2.0	**0.973**	**0.474**	**88.0%**	$K_{ev} = 8$; $K_1 = K_2 = K_3 = 10$
			Pessimistic Performance		
	CLADE	0.859	0.405	41.0%	$K_1 = K_2 = K_3 = 10$
	ftMLDE (EVmutation)	0.850	0.365	35.5%	sampling threshold = 64,000
	ftMLDE (MSA Transformer)	0.860	**0.427**	47.0%	sampling threshold = 1600
	CLADE 2.0	**0.915**	0.425	**58.0%**	$K_{ev} = 2$; $K_1 = K_2 = K_3 = 10$
PhoQ	MLDE (random sampling)	0.387	0.077	7.6%	
			Optimistic Performance		
	CLADE	0.493	0.091	15.0%	K1 = K2 = K3 = 50
	ftMLDE (EVmutation)	0.555	0.103	22.5%	sampling threshold = 9600
	ftMLDE (MSA Transformer)	0.488	0.096	14.5%	sampling threshold = 48,000
	CLADE 2.0	**0.573**	**0.106**	**27.5%**	$K_{ev} = 3$; $K_1 = K_2 = K_3 = 10$
			Pessimistic Performance		
	CLADE	0.445	0.086	9.5%	$K_1 = K_2 = K_3 = 30$
	ftMLDE (EVmutation)	**0.477**	0.101	**13.5%**	Sampling threshold = 6400
	ftMLDE (MSA Transformer)	0.320	0.085	0.0%	Sampling threshold = 1600
	CLADE 2.0	0.425	**0.104**	4.5%	$K_{ev} = 6$; $K_1 = K_2 = K_3 = 40$

[a]Optimistic performance and pessimistic performance are measured by the expected max fitness. To use identical physiochemical encoding for direct comparisons, results for MLDE and ftMLDE were from ref 27. The results from original ftMLDE[4] are only for GB1 set with worse performance than in this table. We reproduced CLADE in this work to have the same set of hyperparameters explored with CLADE 2.0, while the original CLADE work reported slightly better results with extensive hyperparameter search.[27] Each type of ftMLDE only uses one evolutionary score to rank fitness, and the training set is randomly sampled within the top "Sampling threshold" sequences.

CLADE 2.0 uses an iterative process to combine computations and experiments. In each iteration, the computational approach first picks up a few top candidate mutations, and experiment subsequently screens and evaluates the fitness of the selected mutations. In our computational benchmarks, the experimental module is not coupled. When our computational module picks up the candidate mutations, we need to find their fitness values from the existing data set. Indeed, huge combinatorial libraries with almost complete coverage of all mutations are necessary for the computational benchmarks. In applications with available experimental modules, only a small data set is needed to be evaluated by experiments to obtain their fitness. For example, our setting in this work only needs the fitness of 480 mutations over 160,000 in the combinatorial library. Here, 480 mutations are a relatively small portion for experimental data and the number of mutations may be varied depending on the specific problems.

In this work, CLADE 2.0 was only tested computationally on two combinatorial libraries with almost complete coverage of mutations. In practice with experimental module available, CLADE 2.0 can be applied to more general mutational libraries. For example, one can apply CLADE 2.0 to a multidomain protein with more mutational sites involved in a chimera recombinant library, which was studied previously in a MLDE task using Gaussian process model.[23] Unfortunately, there is no existing chimera data set with a complete coverage of mutations for computational benchmark.

The protein design is a complicated process consisting of multiple steps, including discovery, clinical trials, and manufacture.[6] The discovery process also consists of many subprocesses. CLADE 2.0 is responsible for the initial discovery process to select a group of top candidate mutations. Further downstream analysis using both computations and experiments is necessary to screen these top candidates. For example, we need to examine the structural conformation, thermostability, and other critical biophysical properties of the candidate mutations.

DATA AVAILABILITY

The GB1 data set[5] is available at https://www.ncbi.nlm.nih.gov/bioproject/PRJNA278685/ with accession code PRJNA278685. PhoQ data set was reported in the literature.[29] Its processed data used in this work is owned by Michael T. Laub lab and is available at https://github.com/WeilabMSU/CLADE.

CODE AVAILABILITY

All source codes and models are publicly available at https://github.com/WeilabMSU/CLADE-2.0.

ASSOCIATED CONTENT

Supporting Information

The Supporting Information is available free of charge at https://pubs.acs.org/doi/10.1021/acs.jcim.2c01046.

Sequence Identity in CLADE 2.0, physicochemical sequence encoding, MSA, and supervised learning models (PDF)

(PDF)

AUTHOR INFORMATION

Corresponding Author

Guo-Wei Wei − *Department of Mathematics, Michigan State University, East Lansing, Michigan 48824, United States;*

https://doi.org/10.1021/acs.jcim.2c01046
J. Chem. Inf. Model. 2022, 62, 4629−4641

Department of Biochemistry and Molecular Biology and Department of Electrical and Computer Engineering, Michigan State University, East Lansing, Michigan 48824, United States; orcid.org/0000-0002-5781-2937; Email: weig@msu.edu

Author

Yuchi Qiu − *Department of Mathematics, Michigan State University, East Lansing, Michigan 48824, United States*

Complete contact information is available at: https://pubs.acs.org/10.1021/acs.jcim.2c01046

Notes

The authors declare no competing financial interest.

■ ACKNOWLEDGMENTS

This work was supported in part by NIH grants R01GM126189 and R01AI164266, NSF grants DMS-2052983, DMS-1761320, and IIS-1900473, NASA grant 80NSSC21M0023, Michigan Economic Development Corporation, MSU Foundation, Bristol-Myers Squibb 65109, and Pfizer.

■ REFERENCES

(1) Arnold, F. H. Design by directed evolution. *Acc. Chem. Res.* **1998**, *31*, 125−131.
(2) Wu, Z.; Kan, S. J.; Lewis, R. D.; Wittmann, B. J.; Arnold, F. H. Machine learning-assisted directed protein evolution with combinatorial libraries. *Proc. Natl. Acad. Sci.* **2019**, *116*, 8852−8858.
(3) Yang, K. K.; Wu, Z.; Arnold, F. H. Machine-learning-guided directed evolution for protein engineering. *Nat. Methods* **2019**, *16*, 687−694.
(4) Wittmann, B. J.; Yue, Y.; Arnold, F. H. Informed training set design enables efficient machine learning-assisted directed protein evolution. *Cell Syst.* **2021**, *12*, 1026.
(5) Wu, N. C.; Dai, L.; Olson, C. A.; Lloyd-Smith, J. O.; Sun, R. Adaptation in protein fitness landscapes is facilitated by indirect paths. *Elife* **2016**, *5*, No. e16965.
(6) 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.
(7) Gao, K.; Wang, R.; Chen, J.; Cheng, L.; Frishcosy, J.; Huzumi, Y.; Qiu, Y.; Schluckbier, T.; Wei, X.; Wei, G.-W. Methodology-centered review of molecular modeling, simulation, and prediction of SARS-CoV-2. *Chem. Rev.* **2022**, *122*, 11287−11368.
(8) Hsu, C.; Nisonoff, H.; Fannjiang, C.; Listgarten, J. Learning protein fitness models from evolutionary and assay-labeled data. *Nat. Biotechnol.* **2022**, *40*, 1114−1122.
(9) Finn, R. D.; Clements, J.; Eddy, S. R. HMMER web server: interactive sequence similarity searching. *Nucleic Acids Res.* **2011**, *39*, W29−W37.
(10) Hopf, T. A.; Ingraham, J. B.; Poelwijk, F. J.; Schärfe, C. P.; Springer, M.; Sander, C.; Marks, D. S. Mutation effects predicted from sequence co-variation. *Nat. Biotechnol.* **2017**, *35*, 128−135.
(11) 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.
(12) 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. *Adv. Neural Inf. Process Syst.* **2021**, *34*, 29287.
(13) Rives, A.; Meier, J.; Sercu, T.; Goyal, S.; Lin, Z.; Liu, J.; Guo, D.; Ott, M.; Zitnick, C. L.; Ma, J.; Fergus, R. Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences. *Proc. Natl. Acad. Sci.* **2021**, *118*, No. e2016239118.
(14) 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.
(15) Biswas, S.; Khimulya, G.; Alley, E. C.; Esvelt, K. M.; Church, G. M. Low-N protein engineering with data-efficient deep learning. *Nat. Methods* **2021**, *18*, 389−396.
(16) Rao, R.; Liu, J.; Verkuil, R.; Meier, J.; Canny, J. F.; Abbeel, P.; Sercu, T.; Rives, A. Msa transformer. **2021**, bioRxiv:10.1101/2021.02.12.430858.
(17) Wang, M.; Cang, Z.; Wei, G.-W. A topology-based network tree for the prediction of protein−protein binding affinity changes following mutation. *Nat. Mach. Intell.* **2020**, *2*, 116−123.
(18) Cang, Z.; Wei, G.-W. Analysis and prediction of protein folding energy changes upon mutation by element specific persistent homology. *Bioinformatics* **2017**, *33*, 3549−3557.
(19) Nguyen, D. D.; Cang, Z.; Wei, G.-W. A review of mathematical representations of biomolecular data. *Phys. Chem. Chem. Phys.* **2020**, *22*, 4343−4367.
(20) Wittmann, B. J.; Johnston, K. E.; Wu, Z.; Arnold, F. H. Advances in machine learning for directed evolution. *Curr. Opin. Struct. Biol.* **2021**, *69*, 11−18.
(21) Li, W.-D.; Dong, Y.; Reetz, M. T. Can machine learning revolutionize directed evolution of selective enzymes? *Adv. Synth. Catal.* **2019**, *361*, 5069.
(22) Saito, Y.; Oikawa, M.; Nakazawa, H.; Niide, T.; Kameda, T.; Tsuda, K.; Umetsu, M. Machine-learning-guided mutagenesis for directed evolution of fluorescent proteins. *ACS Synth. Biol.* **2018**, *7*, 2014−2022.
(23) Bedbrook, C. N.; Yang, K. K.; Rice, A. J.; Gradinaru, V.; Arnold, F. H. Machine learning to design integral membrane channelrhodopsins for efficient eukaryotic expression and plasma membrane localization. *PLoS Comput. Biol.* **2017**, *13*, No. e1005786.
(24) Romero, P. A.; Krause, A.; Arnold, F. H. Navigating the protein fitness landscape with Gaussian processes. *Proc. Natl. Acad. Sci.* **2013**, *110*, E193−E201.
(25) Mason, D. M.; Friedensohn, S.; Weber, C. R.; Jordi, C.; Wagner, B.; Meng, S.; Gainza, P.; Correia, B. E.; Reddy, S. T. Deep learning enables therapeutic antibody optimization in mammalian cells by deciphering high-dimensional protein sequence space. **2019**, bioRxiv:10.1101/617860.
(26) Hie, B. L.; Yang, K. K. Adaptive machine learning for protein engineering. *Curr. Opin. Struct. Biol.* **2022**, *72*, 145−152.
(27) Qiu, Y.; Hu, J.; Wei, G.-W. Cluster learning-assisted directed evolution. *Nat. Comput. Sci.* **2021**, *1*, 809−818.
(28) Yang, K. K.; Chen, Y.; Lee, A.; Yue, Y. Batched stochastic Bayesian optimization via combinatorial constraints design. *The 22nd International Conference on Artificial Intelligence and Statistics*; PMLR, 2019; pp 3410−3419.
(29) Podgornaia, A. I.; Laub, M. T. Pervasive degeneracy and epistasis in a protein-protein interface. *Science* **2015**, *347*, 673−677.
(30) Hopf, T. A.; et al. The EVcouplings Python framework for coevolutionary sequence analysis. *Bioinformatics* **2019**, *35*, 1582−1584.
(31) Steinegger, M.; Meier, M.; Mirdita, M.; Vöhringer, H.; Haunsberger, S. J.; Söding, J. HH-suite3 for fast remote homology detection and deep protein annotation. *BMC Bioinf.* **2019**, *20*, 473.
(32) Bergstra, J.; Yamins, D.; Cox, D. Making a science of model search: Hyperparameter optimization in hundreds of dimensions for vision architectures. *International Conference on Machine Learning*; PMLR, 2013; pp 115−123.
(33) Pedregosa, F.; et al. Scikit-learn: Machine learning in Python. *J. Mach. Learn. Res.* **2011**, *12*, 2825−2830.
(34) Ketkar, N. Introduction to Keras. *Deep Learning with Python*; Springer, 2017; pp 97−111.
(35) Chen, T.; Guestrin, C. Xgboost: A scalable tree boosting system. *Proceedings of the 22nd acm Sigkdd International Conference on Knowledge Discovery and Data Mining*; Association for Computing Machinery, 2016; pp 785−794.

(36) Hamerly, G.; Elkan, C. Learning the k in k-means. *Adv. Neural Inf. Process. Syst.* **2004**, *16*, 281−288.
(37) Kawashima, S.; Ogata, H.; Kanehisa, M. AAindex: amino acid index database. *Nucleic Acids Res.* **1999**, *27*, 368−369.
(38) Zamyatnin, A. Protein volume in solution. *Prog. Biophys. Mol. Biol.* **1972**, *24*, 107−123.
(39) Ofer, D.; Linial, M. ProFET: Feature engineering captures high-level protein functions. *Bioinformatics* **2015**, *31*, 3429−3436.
(40) Georgiev, A. G. Interpretable numerical descriptors of amino acid space. *J. Comput. Biol.* **2009**, *16*, 703−723.

https://doi.org/10.1021/acs.jcim.2c01046
J. Chem. Inf. Model. 2022, 62, 4629−4641