CLADE: Cluster learning-assisted directed evolution

Yuchi Qiu
Michigan State University
Jian Hu
Michigan State University
Guo-Wei Wei (weig@msu.edu)
Michigan State University

Article

Keywords: Protein engineering, directed evolution, machine learning, clustering, fitness

Posted Date: June 7th, 2021

DOI: https://doi.org/10.21203/rs.3.rs-528258/v1

Version of Record: A version of this preprint was published at Nature Computational Science on December 9th, 2021. See the published version at https://doi.org/10.1038/s43588-021-00168-y.

CLADE: Cluster learning-assisted directed evolution

Yuchi Qiu[1], Jian Hu[2,3] and Guo-Wei Wei [*1,3,4]

[1]*Department of Mathematics, Michigan State University, East Lansing, MI 48824, USA*
[2]*Department of Chemistry, Michigan State University, MI, 48824, USA*
[3]*Department of Biochemistry and Molecular Biology, Michigan State University, MI, 48824, USA*
[4]*Department of Electrical and Computer Engineering, Michigan State University, MI 48824, USA*

Abstract

Directed evolution (DE), a strategy for protein engineering, optimizes protein properties (i.e. fitness) by expensive and time-consuming screen or selection of a large combinatorial sequence space. Machine learning-assisted directed evolution (MLDE) that screens variant properties *in silico* can reduce the experimental burden. However, the MLDE utilizing small experimentally labeled training data from random sampling renders low global maximal fitness hitting rates. This work introduces a cluster learning-assisted directed evolution (CLADE) framework, particularly designed for systems without high-throughput screening assays, that combines sampling through hierarchical unsupervised clustering and supervised learning to guide protein engineering. Based on general biological information, CLADE splits the genetic combinatorial space into various subspaces with heterogeneous evolutionary traits, which guides the selection of experimental sampling sets and the subsequent building up of supervised learning training sets. By virtually screening two four-site combinatorial fitness landscapes from protein G domain B1 (GB1) and PhoQ, our CLADE consistently showed near 3-fold improvement on global maximal fitness hitting rate than using randomly sampled training data. Our CLADE can be easily applied to various biological systems and customized for systems with different throughput levels to maximize its accuracy and efficiency. It promises a significant impact to protein engineering.

Key words: Protein engineering, directed evolution, machine learning, clustering, fitness

Contents

*Corresponding author: weig@msu.edu

1 Introduction

Directed evolution (DE) is a commonly used approach in protein engineering to improve certain properties (e.g., fitness) of a target protein. The fitness landscape is a high-dimensional surface that maps amino acid sequences to properties including activity, selectivity, stability, and other physicochemical features. Conventional DE seeks to discover useful variants satisfying desired properties by searching the optimal sequences on the fitness landscape through selection or screen. However, the full exploration of the fitness landscape is difficult under restricted timelines and laboratory capacities particularly when a high-throughput selection or screen is not available for the system because the size of the sequence space is in the order of 20^L with L potential amino acids to be changed [1].

The last decade has witnessed the rapid development of machine learning and deep learning algorithms for biological data [2, 3, 4, 5, 6]. Supervised models can learn relationships between sequences and fitness properties, and provide quantitative predictions on protein thermostability [7], protein folding energy [8, 9], protein solubility [10], protein-ligand binding affinity [11], and protein-protein binding affinity [12]. Due to the high cost of acquiring supervised protein labels, self-supervised protein embedding has emerged as an important paradigm in protein modeling. Trained on vast unlabeled sequence data resulting from natural evolution, self-supervised protein embedding can capture significant latent biological information of sequence and pass the information to the downstream supervised task [13, 14]. Adapted from natural language processing, many model architectures, such as variational auto-encoder [15], recurrent neural network [16, 17], and transformer [18], can be used to train the protein embedding models [13]. On the other hand, unsupervised clustering methods can identify the internal characteristics of unlabeled data by dividing them into multiple subspaces. Clustering methods, including distance-based clustering [19, 20], community-based clustering [21], density-based clustering [22], and graph-based clustering [23, 24], were widely applied to transcriptomic data analysis [25], pattern recognition [26] and image processing [27] to reveal data heterogeneity.

DE optimizes protein fitness by mimicking the process of natural selection [28]. The epistasis is prevalent in the fitness landscape, where the combined effect of multiple mutations deviates from that predicted by adding their individual effects [29]. The DE via single-mutation search is generally restricted to exploring local valleys due to the epistasis [30, 31, 32], whereas multi-site saturation mutagenesis is inevitably associated with a huge combinatorial library, which often overwhelms the screen capacity. Recently, machine learning-assisted directed evolution (MLDE) becomes a new approach to navigate the epistatic fitness landscape for a predetermined combinatorial library at selected mutation sites. In MLDE, a supervised learning model is trained on a small sample of experimentally labeled variants ($\sim 10^2$) and is used to predict the fitness of all the unlabeled variants in the combinatorial library. Variants with top predicted fitness are experimentally screened to find optimal variants [1, 33, 34]. The MLDE has been applied to improve protein fitness in numerous biological systems, such as enzyme evolution [31], engineering of GFP fluorescence [35], the localization of membrane proteins [36], protein thermostability optimization [37], therapeutic antibody optimization [38].

Functional proteins are rare in the enormous combinatorial space, and as the desired level of function increases the number of variants having that function decreases exponentially [1]. It is challenging for the MLDE to accurately predict high-fitness variants by learning from the training data overwhelmed with low- or zero-fitness variants. The application of zero-shot prediction, which predicts protein functions without any data collection, can be an effective approach in selecting more informative variants in the training data. With the inclusion of the zero-shot predictor, the focused training MLDE achieved significant improvement

in predicting fitness landscape comparing to traditional DE on protein G domain B1 (GB1) dataset [30]. However, the unsupervised zero-shot predictor requires large amounts of prior knowledge in predicting a property for all variants and this property needs to be highly correlated with the desired fitness. The generalization of the zero-shot predictor is difficult and intricate where customized designs and testing are necessary before application to a new biological system or a new type of protein fitness.

In this work, we purpose a novel cluster learning-assisted directed evolution (CLADE) framework to guide protein engineering. CLADE framework introduces an unsupervised clustering strategy to preselect the training sets for supervised learning to virtually navigate the fitness landscape. Through unsupervised clustering methods, the fitness heterogeneity can be identified where clusters have significantly different populations of high-fitness variants. Utilizing the fitness heterogeneity, we identify and oversample the clusters enriched with high-fitness variants according to the cluster-wise sampling probability which is dynamically updated and iterated with experimental screen. By introducing a hierarchical structure in clustering method, the performance of CLADE is accurate and robust with respect to the selection of hyperparameters. With the requirement of the same amount of prior knowledge with MLDE, CLADE can reach 50.8% and 55.8% global maximal fitness hitting rates for simulated medium and low throughput systems, respectively, which are over 2.7-fold improvement to MLDE on GB1 dataset. We further tested CLADE on the PhoQ dataset whose fitness is more sophisticated and rare than GB1 [39] and a 2.9-fold improvement on global maximal fitness hitting rate (i.e. from 7.2% to 20.6%) can be found comparing to MLDE. Our CLADE can be easily customized to systems with various throughput levels and particularly, low throughput systems may be more beneficial to achieve higher global maximal fitness hitting rate.

2 Results

2.1 Overview of CLADE

The CLADE framework consists of the experimental screen, unsupervised clustering, and supervised learning, where unsupervised clustering and supervised learning serve as complementary roles to guide experimental screen to discover variants with optimal fitness in directed evolution (Figure 1A). Prior to CLADE, a target protein and multiple sites for saturation mutagenesis need to be determined by expert selection. An unlabeled combinatorial library is then constructed which consists of sequences of all candidate variants (Figure 1B). The unknown specific fitness information can be determined through the experimental screen, but usually only a small subset of variants is screened due to experimental constraints. Although specific fitness information is largely unknown, general biological information, such as amino acid physicochemical property, is available for all variants in the combinatorial library (Figure 1B). A hidden correlation between general biological information and specific fitness information variants can be learned. At the first stage of CLADE, unsupervised clustering guides coarse search and selection over clusters. By encoding sequences of variants with general biological information, unsupervised clustering divides the combinatorial library into multiple clusters with different internal characteristics. Variants in the same cluster have similar general biological properties, as well as fitness properties of the interest despite their values are unknown. Instead of a random selection of variants in the entire combinatorial library, CLADE selects variants via a cluster-learning sampling approach. To select one variant, one cluster is first selected according to the predefined cluster-wise sampling probabilities, and an uniform and random sampling selects the variant in this cluster. The selected variants are experimentally screened to obtain their fitness values. The overall fitness property of each cluster can be approximated by the average fitness of all selected variants in this cluster. The cluster-learning sampling iteratively selects variants and updates the cluster-wise sampling probabilities based on the overall fitness property over clusters. The labeled sample set is taken as training data to train a supervised learning model and provide a quantitative virtual evaluation of the rest of the combinatorial library. Top predicted variants are screened by experiments to discover the optimal variants and evaluate the predictive performance of CLADE (Figure 1C).

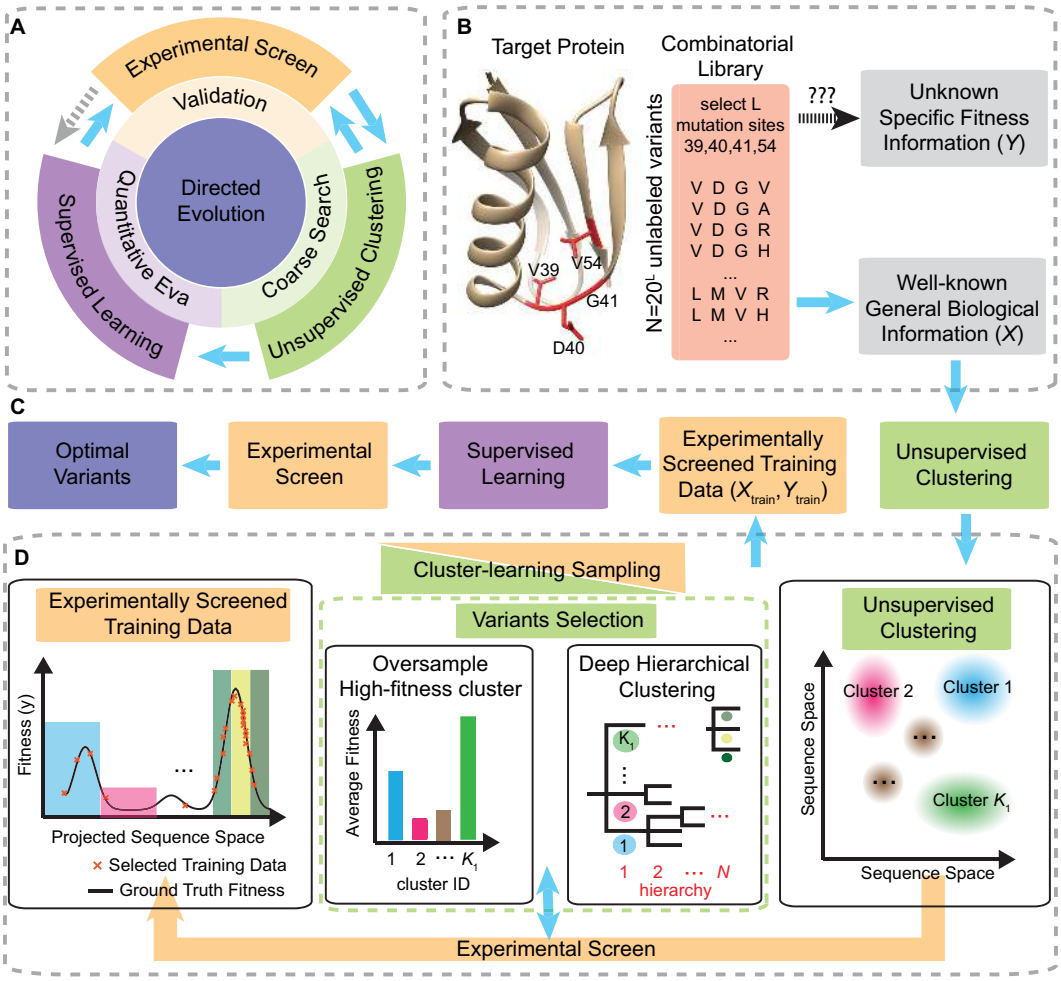

Figure 1: Overview of cluster learning-assisted directed evolution (CLADE). (A) Conceptual diagram of CLADE. CLADE consists of three components: experimental screen, unsupervised clustering, and supervised learning. Unsupervised clustering guides a cluster-wise coarse search of variants and selection by iterating with the experimental screen. The information obtained by unsupervised learning is passed to supervised learning for quantitative evaluation. The experimental screen provides validation of the quantitative evaluation. Blue arrows illustrate the flow of information. The supervised learning can also be repeated after experimental validation, but it is not considered in this work (gray dash arrow). **(B)** Combinatorial library construction. For a target protein, expert selection picks L sites for saturation mutagenesis to construct the combinatorial library including all variants at these sites. In the figure, target protein is GB1 (PDB ID: 2gi9) and $L = 4$ mutation sites are V39, D40, G41, and V54. Each variant can be encoded by well-known general biological information and the encoding of the combinatorial library leads to a feature matrix X. The specific fitness information for each variant is unknown and the experimental screen is required to obtain the precise fitness value, but usually only a small subset of variants can be screened with limited experimental capacity. **(C)** Flowchart of CLADE. Unsupervised clustering divides the combinatorial library into multiple clusters by using the feature matrix X. Cluster-learning sampling selects and screen variants to construct a labeled sample set through iterations between the experimental screen and unsupervised clustering. The labeled sample set is taken as training data passing to the supervised learning. Supervised learning learns from the training data and provides predictions on optimal variants. **(D)** Cluster-learning sampling schematic diagram. Cluster-wise sampling probabilities guide variants selection and the follow-up experimental screen at different clusters. Sampling probabilities are calculated based on existing labeled variants and dynamically updated when a new batch of variants is screened. Clusters with high average fitness tend to be oversampled with higher sampling probabilities. Deep hierarchical clustering is calculated during iterations to further oversample the high-fitness clusters. The high-fitness clusters are divided into more subclusters to allow further oversampling in these clusters.

In cluster-learning sampling, cluster-wise sampling probabilities are dynamically updated after each batch of variants is screened (Figure 1D). In the first few batches, sampling probabilities are identical for all clusters to have a rough coverage of all clusters. Then the sampling strategy is designed to oversample the high-fitness clusters since high-fitness variants are more desired in fitness optimization. The sampling probability for each cluster is defined by the average fitness of selected variants in this cluster divided by the summation of the average fitness of selected variants in each cluster (Methods). To further oversample the high-fitness clusters, we propose a deep hierarchical clustering structure (Figure 1D). Clusters with higher average fitness are divided into more subclusters and then the same cluster-wise sampling procedure is applied to clusters at the new hierarchy. For maximum hierarchy N, N hyperparameters are needed for the increment of new clusters at each hierarchy. The increment of clusters at hierarchy i is defined as K_i ($i = 1,\ 2,\ \cdots,\ N$) (Methods). Three examples of simulated sampling with various maximum hierarchies were presented to further illustrate the sampling process (Supplementary Information S3, Figure S1).

In the experimental screen, a batch of variants is usually screened in parallel and the batch size varies in systems with different throughput systems. To adopt CLADE to systems with different throughputs, the frequency for updating sampling probability or generating clusters at new hierarchy needs to be multiples of the batch size, as well as the number of training data and the number of top-predicted variants being screened. Two batch sizes, 96 and 1, were taken in this work. Batch size 96 is followed by the small 96-well plate commonly seen in many experimental systems [31, 35] and it is used to simulate medium throughput systems. Batch size 1 is used to simulate systems with extremely low throughput where variants need to be screened one by one. For these two types of systems, many procedures are identical in this work: 1) the size of training data is 384 and top 96 predicted variants are screened to evaluate the predictive performance; 2) the first 96 samples are selected randomly and uniformly over clusters; 3) new subclusters at new hierarchy are generated after every 96 variants are collected until reaching the maximum hierarchy N. The only difference is the frequency for updating sampling probabilities, which is identical to the batch size. The outcome of CLADE consists of variants in the training data and the top 96 predicted variants. The max fitness and mean fitness are used to evaluate the CLADE outcome. Another important metric, the global maximal fitness hitting rate, measures the frequency that CLADE successfully picks the global maximal variant in either training data or top prediction. Details and more metrics are given in Methods.

2.2 Unsupervised clustering reveals fitness heterogeneity

The fitness landscape is usually enriched with low- or zero-fitness variants [1]. For example, an empirically determined combinatorial fitness landscape of protein G domain B1 (GB1; PDB ID: 2gi9) consists of experimentally determined fitness [32]. The fitness was defined as the enrichment of folded protein bound to the antibody IgG-Fc. This data set contains 149,361 variants out of $20^4 = 160{,}000$ variants at four amino acid sites (V39, D40, G41, and V54). By normalizing the fitness to its global maximum, 92% of variants have fitness lower than 0.01 and 99.3% variants have fitness lower than 0.3. As a case study, we tested our CLADE method on the GB1 dataset.

As a proof of principle, we employed K-means clustering and took four physicochemical descriptors, AA encoding, as the sequence encoding method (Methods). We first cluster the fitness landscape into $K_1 = 3$ clusters. Three clusters contain the similar number of variants and they are well separated in the projected principal components space. The population of high-fitness variants (i.e. > 0.3) is rare in the fitness landscape. Interestingly, the heterogeneity of high-fitness variants was found in these clusters, where cluster 3 contains over 11-fold of high-fitness variants than either cluster 1 or cluster 2 (Figure 2A).

Next, we performed the K-means clustering with various numbers of clusters K_1 (10, 40, and 100) and multiple repeats were performed for each K_1 value. In a single simulation, clusters were numbered by a unique cluster ID, where cluster ID indicates the descending ranking of the average fitness for all variants within the corresponding cluster. Expected average fitness in the cluster with identical cluster ID were calculated (Figure 2B). The distribution of cluster average fitness reveals the fitness heterogeneity where the

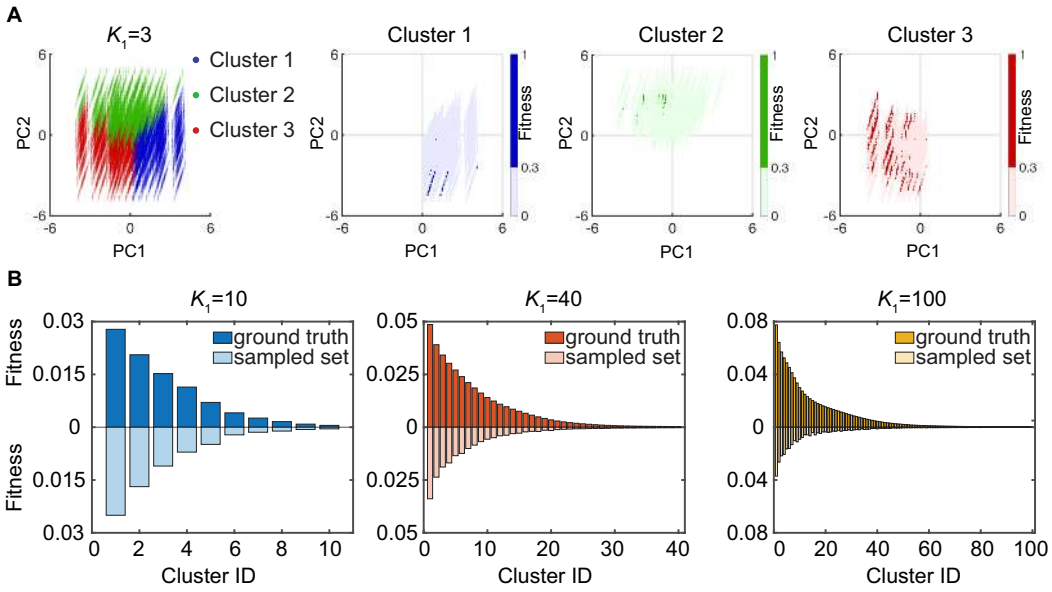

Figure 2: K-means reveals fitness heterogeneity and cluster-learning sampling recapitulates the heterogeneity with maximum hierarchy $N = 1$**. (A)** Visualization of GB1 variants in the reduced two-dimensional space spanned by the first two principal components. Three clusters were obtained from K-means. Dots with different colors represent variants in different clusters. Each cluster was plotted individually (from the second subplot to the fourth subplot). Variants with fitness lower or higher than 0.3 are denoted by light or dark colors, respectively. Numbers of variants in three clusters are 50,030, 51,016, and 48,315, respectively. And numbers of high-fitness variants (i.e. > 0.3) in these clusters are 80, 59, and 911, respectively. **(B)** K-means clustering and the follow-up cluster-learning sampling on the GB1 dataset with 500 independent repeats. Three sets of parameters are presented individually in different plots: $K_1 = 10$ (blue), 40 (red), and 100 (yellow). In a single simulation, each cluster is numbered by a unique cluster ID, where cluster ID indicates the descending ranking of the average fitness for all variants within the corresponding cluster. Bar plots above the abscissa with dark color show the expected average ground-truth fitness for all variants contained in each cluster. Bar plots below the abscissa with light color show the expected average fitness for variants selected from the cluster-learning sampling in each cluster.

cluster with lower numbering has higher average fitness (Figure 2B). We found the distribution of cluster average fitness becomes more polarized near the origin as K_1 increases. Specifically, 32%, 52% and 67% of high-fitness variants (i.e. > 0.3) are contained in the top 10% clusters for K_1 values at 10, 40, and 100, respectively (Figure 2B).

The cluster-learning then oversampled the high-fitness cluster in the simulated medium-throughput system. In sampled data, distributions of the expected cluster average fitness recapitulated the polarized distributions as shown in the ground-truth fitness and the distributions become more polarized as K_1 increases (Figure 2B). Indeed, K-means can capture the fitness heterogeneity and our cluster-learning algorithm can recapitulate this heterogeneity and select more samples with higher fitness. A community-based clustering method, Louvain clustering [21], was also carried out to capture the fitness heterogeneity (Supplementary Information S4, Figure S2).

2.3 Accurate and robust CLADE outcome with deep hierarchical structure

Utilizing the fitness heterogeneity, CLADE performed differently under different clustering architectures. First, we performed CLADE on simulated medium-throughput systems by exploring maximum hierarchy N and hyperparameters (i.e. increments of clusters at each hierarchy). With shallow hierarchy $N = 1$, CLADE using K-means improved all evaluating metrics, including expected max fitness, expected mean fitness, global maximal hitting rate, NDGC, cross validation errors, and testing errors, for both training data and

top 96 predicted variants, comparing to the case using the random sampled training data regardless of the parameters selection (Table S1-S3). Moreover, the global maximal fitness hitting rate can reach 40.2% when $K_1 = 90$, a 2.2-fold improvement to the case using the random sampled training data (Table 1). Similarly, by exploring hyperparameters of Louvain method, CLADE can lead to similar improvement and an almost 2-fold improvement on global maximal fitness hitting rate, 36.4%, can be observed (Table 1). In hierarchical clustering, a cluster may contain fewer variants than the number of its subclusters at the next hierarchy since the number of variants in one cluster decreases quickly with respect to its hierarchy. To avoid this issue, various cluster increments (K_1, K_2, K_3, etc.) are explored in smaller ranges for deep hierarchy. With deep hierarchy, CLADE performance was further improved (Table S1-S3). A 2.7-fold improvement of the global maximal hitting rate, 50.8%, can be observed for both $N = 2$ and $N = 3$ (Table 1).

Clustering method; architecture	Parameters	Expected max fitness	Expected mean fitness	Global max hitting rate
random sampling (MLDE); $N = 0$;	–	0.774	0.305	18.6%
K-means; $N = 1$	$K_1 = 90$	0.870	0.406	40.2%
Seurat (Louvain); $N = 1$	k.param=500; resolution=1.2	0.846	0.357	36.4%
K-means; $N = 2$	$K_1 = 40$; $K_2 = 30$	0.887	0.421	50.8%
K-means; $N = 3$	$K_1 = 30$; $K_2 = K_3 = 40$	**0.888**	**0.423**	**50.8%**
Low throughput; K-means; $N = 3$	$K_1 = 30$; $K_2 = K_3 = 50$	**0.904**	**0.431**	**55.6%**

Table 1: CLADE performance GB1 dataset with different sampling architectures by using AA encoding. For each architecture, hyperparameters for clustering method were explored (Table S1-S3). The case with highest expected max fitness for each architecture was shown in this table. Unless explicitly indicated, the batch size is taken as 96 to simulate the medium-throughput systems. The case with $N = 0$ indicates randomly sampled training data which is equivalent to the MLDE approach. All statistics were obtained from 500 independent repeats of both sampling and training. Expected max fitness and expected mean fitness were evaluated on top 96 variants from supervised learning model. The global maximum hitting rate was evaluated on the union of the top 96 variants from supervised learning model and the 384 variants in training data.

In applications, the robustness of CLADE performance to hyperparameters is also desired since only one set of hyperparameters can be picked and applied. Surprisingly, the robustness was enhanced as the maximum hierarchy increases (Figure S3-S5, Table S1). With shallow hierarchy $N = 1$, the global maximal fitness hitting rate is relatively low and varies in a relatively large range from 30.6% to 41.2%. While for deep hierarchy $N = 3$, the global maximal fitness hitting rate is relatively higher and varies in a relatively small range from 41.6% to 50.8%.

We also performed CLADE in the simulated low-throughput systems. We only explored CLADE with maximum hierarchy $N = 3$, which achieves the best performance in medium-throughput systems. Because sampling probabilities are updated more frequently, the simulated low-throughput systems can achieve better performance measured in expected max fitness, expected mean fitness, and global maximal fitness hitting rate. Especially, the global maximal fitness hitting rate can reach 55.6% (Table 1).

Overall, deep CLADE ensures robust and accurate performance in directed evolution. Systems with lower throughput may achieve better performance.

2.4 CLADE mediates training data diversity to improve its outcome

In CLADE, various clustering architectures result in different compositions of training data and affect the outcome of the downstream supervised learning. Training data diversity is critical to the outcome of the supervised learning model, where high diversity may minimize the extrapolation and low diversity may allow more accurate predictions at a local structure. Here, we study training data diversity in both feature space (i.e. sequence diversity) and labels space (i.e. fitness diversity), and both of them are quantified by the modified functional attribute diversity (MFAD) (Methods).

We compared four CLADE simulations with various maximum hierarchies N from 0 to 3 on GB1 dataset with AA encoding, particularly, $N = 0$ represents random sampling without clustering (Figure 3A-F). We picked increments of clusters such that any cases at the same hierarchy have the same number of clusters despite their different maximum hierarchy. All cases are overwhelmed with low- or zero-fitness, which is inherent from the fitness landscape. But as the maximum hierarchy N increases, more high-fitness variants can be selected and the fitness distribution becomes less localized at 0, especially for the last batch of selection where variants were selected at the maximum hierarchy (Figure 3A and Figure S6). As a result, fitness diversity increases when a new hierarchy is added. On the other hand, we observed distributions of variants in reduced sequence space become more localized as a new hierarchy is added, as a result, the sequence diversity decreases (Figure 3C-F and Figure S7). With increased fitness diversity and reduced sequence diversity in training data, the performance of the downstream supervised learning model is improved for both max fitness and mean fitness of top predicted variants with deep hierarchy (Figure 3B). The consistent conclusion can be drawn statistically with multiple repeats (Figure S8).

By aligning statistical results from different CLADE architectures and hyperparameters, relations between training data diversity and CLADE outcome can be clearly seen (Figure 3G-I). In general, a deeper hierarchy results in lower sequence diversity and higher fitness diversity (Figure 3G). Although shallow hierarchy can reach the similar fitness diversity level with the deep hierarchy with sufficient large K_1, it has much higher sequence diversity close to that in random sampling (Figure 3G). CLADE generally achieved better performance on expected max fitness if lower sequence diversity is achieved in training data (Figure 3H). On the other hand, with increasing fitness diversity, shallow CLADE performance can be improved first with small K_1 but then drop with large K_1. In contrast, deep CLADE performance continues to be improved with increasing fitness diversity (Figure 3I). Such improvement from deep CLADE is not limited to expected max fitness but all evaluating metrics we discussed in this work, including expected mean fitness, global maximal fitness hitting rate, NDCG, cross validation errors, and testing errors (Table S1-S3). Overall, the deeper maximum hierarchy allows further improvement of CLADE performance through mediation on the training data diversity.

2.5 CLADE on PhoQ dataset

We further tested CLADE on PhoQ dataset, a fitness landscape on a combinatorial library with four mutation sites (A284, V285, S288, and T289) [39]. This data set consists of 140,517 labeled variants out of 20^4=160,000. The fitness was defined as the enrichment of similar phosphatase activity with wild-type PhoQ to its substrate PhoP. By normalizing the fitness to its global maximum, PhoQ dataset was found to be overwhelmed with low- or zero-fitness variants with 92% of variants having fitness lower than 0.01 and 99.96% of variants having fitness lower than 0.3, where the high-fitness variants are more rare than that in GB1 dataset (Figure S9A).

Unlike GB1 where the value of fitness directly reflects the levels of protein property (i.e. binding affinity), the value of fitness for PhoQ is more sophisticated as it does not indicate the level of protein property. Using AA encoding, we found a 19% improvement on max fitness and a 3-fold improvement on global maximum

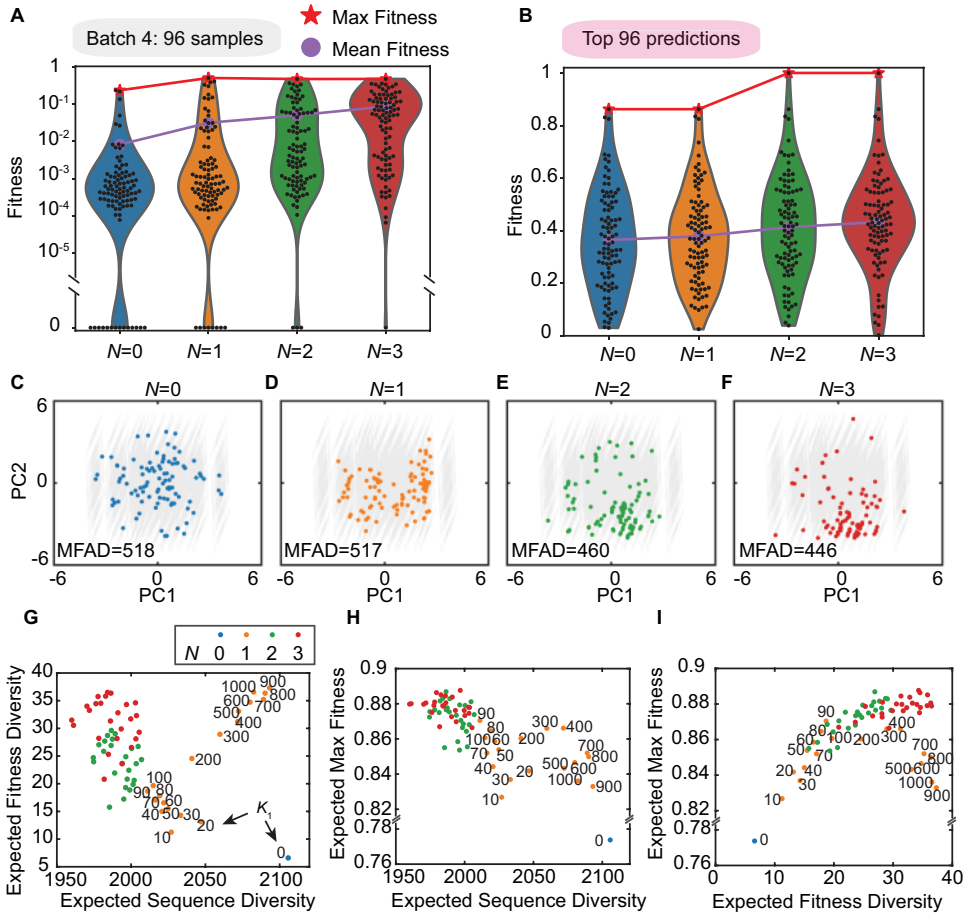

Figure 3: Relations between training data diversity and CLADE outcome. (A-F) Single CLADE simulation with various maximum hierarchies: 1) $N = 0$ (random sampling; MLDE); 2) $N = 1$ ($K_1 = 30$); 3) $N = 2$ ($K_1 = K_2 = 30$); 4) $N = 3$ ($K_1 = K_2 = K_3 = 30$). Distributions of fitness in **(A)** training data sampled at fourth batch (consists of 96 samples) and **(B)** the top 96 predicted variants. The violin plot outlines illustrate kernel probability density where the width of the shaded area represents the proportion of the data located there. Each black dot represents a variant and its ordinate show the fitness of the variant. The red line shows the maximum fitness and the purple line shows the mean fitness. In **(A)**, fitness diversity measured by modified functional attribute diversity (MFAD) are: 1) $N = 0$: 1.4; 2) $N = 1$: 5.3; 3) $N = 2$: 7.4; and 4) $N = 3$: 10.2. Distributions of variants selected at fourth batch in sequence space in the projected first two-principle component space: **(C)** $N = 0$; **(D)** $N = 1$; **(E)** $N = 2$; and **(F)** $N = 3$. The sequence diversity measured by MFAD are: 518, 517, 460, and 446, respectively, for these four cases. In **(C-F)**, gray dots show all variants in the combinatorial library. **(G-H)** For various maximum hierarchies, hyperparameters were explored (Table S1-S3). For $N = 1$, two ranges of K_1 were explored: 10:10:90 and 100:200:1000. For $N = 2$, combinations of K_1 and K_2 were explored: $K_1 = 10 : 10 : 50$ and $K_2 = 10 : 10 : 50$. For $N = 3$, K_3 was assumed to be identical to K_2. Combinations of K_1 and K_2 were explored: $K_1 = 10 : 10 : 50$ and $K_2 = 10 : 10 : 50$. For each set of hyperparameters, CLADE was repeated independently 500 times and expected values of training data fitness diversity, training data sequence diversity, and expected maximum fitness from CLADE are shown. Numbers next to dots inside the plots for cases $N = 0$ or $N = 1$ denote the number of clusters at the first hierarchy, K_1. **(G)** Expected sequence diversity versus expected fitness diversity. **(H)** Expected sequence diversity versus expected maximum fitness from CLADE. **(I)** Expected fitness diversity versus expected maximum fitness from CLADE.

hitting rate from deep CLADE comparing to CLADE using randomly sampled training data (i.e. MLDE). However, the improved predictive performance from deep CLADE still has low expected max fitness and low global maximal fitness hitting rate. Instead of using AA encoding extracted from a small subset of AAIndex [40], Georgiev encoding [41, 42], a more comprehensive encoding method by integrating over 500

amino acid indices in AAIndex database, was tested. We found CLADE performance was largely improved by using Georgiev encoding. Deep CLADE again showed significant improvement compared to the case uses randomly sampled training data, where a 36% improvement on expected max fitness and a 2.9-fold improvement (i.e. from 7.2% to 20.6%) on global maximum hitting were observed (Table 2). Despite of CLADE showing lower global maximum hitting rate and expected max fitness in PhoQ dataset than that in GB1 dataset, the fitness improvement relative to wild-type protein measured by expected max fitness is much higher for PhoQ, which are 7.8- and 67-fold, respectively, for GB1 and PhoQ (Figure S9B).

Encoding method	Architecture	Expected max fitness	Expected mean fitness	Global max hitting rate
AA	random sampling (MLDE); $N = 0$	0.299	0.072	1.0%
AA	$N = 3$; $K_1 = 40$; $K_2 = K_3 = 30$	0.357	0.093	3.0%
Georgiev	random sampling (MLDE); $N = 0$	0.371	0.077	7.2%
Georgiev	$N = 3$; $K_1 = 10$; $K_2 = K_3 = 30$	**0.503**	**0.096**	**20.6%**

Table 2: CLADE performance with different encoding methods on PhoQ data set. Deep CLADE was only explored for maximum hierarchy $N = 3$ and parameters were explored (Table S1-S3). All cases used K-means for clustering method. The cases with highest expected max fitness were shown in this table. Unless explicitly indicated, the batch size is taken as 96 to simulate the medium-throughput systems. The case with $N = 0$ indicates randomly sampled training data which is equivalent to the MLDE approach. All statistics were obtained from 500 independent repeats including sampling and training. Expected max fitness and expected mean fitness were evaluated on top 96 variants from supervised learning model. The global maximum hitting rate was evaluated on the union of the top 96 variants from supervised learning model and the 384 variants in training data.

3 Discussions

In this study, we proposed a cluster learning-assisted directed evolution framework. CLADE is effective to identify the heterogeneity of the fitness landscape by utilizing general biological information. Then, the cluster-learning sampling is able to recapitulate such heterogeneity to provide more informative training data. With the proposed deep hierarchical structure in clustering, we found CLADE is efficient to assist experiment to find high-fitness variants and its performance is robust to the selection of hyperparameters. In applications, after expert selection on potential mutation sites for saturation mutagenesis, CLADE can efficiently navigate the fitness landscape *in silico* by selecting and learning from a small subset of experimentally screened variants. It requires only general biological information such as amino acid physiochemical properties, but no specific information on fitness or target protein. CLADE can simply be customized to different biological systems to maximize its impacts. Especially, low-throughput systems may benefit more from this framework.

In general, fitness diversity also reflects the enrichment of high-fitness in the training data for the fitness landscape containing a large portion of low- or zero-fitness variants. Increased fitness diversity allows CLADE to learn more information from high-fitness variants. Deep hierarchical structure in CLADE significantly improves fitness diversity. Because the epistatic landscape has multiple local optima, variants may scatter over multiple local optima. Increased fitness diversity may not ensure CLADE improvement when it exceeds a certain level (Figure S3). To further improve CLADE performance, training data with more variants

near the global optima may help (i.e., reduced sequence diversity). While extremely low sequence diversity may have opposite effects on supervised learning [30]. In deep CLADE, sequential selection of variants over clusters from shallow to deep hierarchies can obtain both low and high sequence diversity at different batches (Figure S7). Therefore, deep CLADE properly regulates training data diversity properly to improve its performance.

CLADE can be implemented by using any sequence encoding methods. In this work, two physicochemical sequence encoding methods were tested. Regardless of the encoding method or the dataset, deep CLADE consistently showed near 3-fold improvement on global maximal fitness hitting rate comparing to CLADE using randomly sampled training data (i.e. MLDE) (Table S4). Interestingly, CLADE on GB1 using AA encoding has better predictive performance than that using Georgiev encoding, while PhoQ behaved other way around. AA encoding is a subset of AAIndex while Georgiev gives a comprehensive low-dimensional representation of AAIndex. For the GB1 dataset, the AA encoding may be sufficient to learn the fitness, and Georgiev encoding may contain redundant information leading to its underperformance than AA encoding. For the PhoQ dataset, due to its sophisticated fitness property, four physicochemical descriptors from AA encoding may not be sufficient to learn the fitness, consequently, Georgiev encoding outperforms AA encoding. To maximize the impacts of CLADE, a universal and informative encoding method is desired. The physicochemical descriptors have been widely applied to many other machine learning tasks in predicting protein functions [8, 12, 43]. Moreover, the development of self-supervised pretraining methods provides novel data-driven approaches in sequence encoding methods [13, 44]. While they were reported to underperform Georgiev encoding on GB1 dataset in MLDE [30], the self-supervised learning enriched with hidden information should be further explored. A careful design for the target protein may be necessary. For example, more homology of the targeted protein can be included in the training data of the self-supervised learning model [33].

The fitness landscape is usually overwhelmed with low- or zero-fitness variants. Avoiding the non-functional variants in training data would significantly improve directed evolution performance. The zero-shot predictor, utilizing large amount of prior knowledge to predict whether a variant is functional, can guide the training data creation to exclude low- or zero-fitness variants. The zero-shot predictor requires its predicted quantities highly correlated to the fitness of interest. The focused training MLDE (ftMLDE) combining a zero-shot predictor with MLDE performed extremely well on the GB1 dataset with 92% global maximal fitness hitting rate [30]. However, the design of the zero-shot predictor may vary much from different proteins and the target fitness properties. The performance of such an unsupervised zero-shot predictor is difficult to be tested before applications. Alternatively, the selection of every batch of variants in CLADE is simply driven by the previously screened variants. Our CLADE provides a general framework in directed evolution that significantly improves the performance of traditional MLDE without the requirement of extra prior knowledge. Although CLADE achieved lower global maximum hitting rate (i.e. 50.8%) on GB1 dataset than ftMLDE, we showed the generalization to more sophisticated fitness on the PhoQ data set, where the value of fitness no longer represents levels of certain protein property and fewer variants near the global maximal fitness (Figure S9). Moreover, integrating additional partial prior knowledge with the CLADE framework, similar to the zero-shot predictor, may further improve CLADE performance.

Iterating with experimental screen, our cluster-learning sampling approach is a special type of active learning in protein engineering [3]. The current active learning methods usually use supervised learning to make decisions for the next round of experiment. Followed by the supervised learning at the end, the CLADE may significantly enhance the outcome robustness by exploring more diverse space and exclusion of low- or zero-fitness variants while preserving sequence and fitness diversity in the training set. In contrast to the current MLDE protocols where site-directed mutagenesis is conducted to generate the variants used to train ML models, the CLADE protocol requires making specific variants throughout the whole process from the initial sampling and the training sets to the predicted set, which would increase experimental cost in making constructs. However, with the rapid decrease in the cost of gene synthesis and development of high-throughput site-directed mutagenesis [45], making hundreds of variants harboring multiple mutations would

still be efficient and affordable. The increased cost would be sufficiently compensated by the significantly improved performance of the supervised learning with the increased expected max fitness and global max hitting rate.

4 Methods

Physicochemical sequence encoding In this work, two types of physicochemical sequence encoding methods, AA and Georgiev, were used to test CLADE. The encoding matrix of the combinatorial library was standardized via *StandardScalar()* in scikit-learn [46] before further usage. The same encoding matrix was used for both unsupervised clustering and supervised learning models. First, the AA encoding consists of four physicochemical descriptors including molecular mass, hydropathy, surface area, and volume (Table S5). Molecular mass, hydropathy, and surface area are obtained from the AAIndex database [40], and volume is from the experimental work [47]. This encoding was previously used in protein stability changes predictions [8]. Instead of picking a subset of AAIndex database, the Georgiev encoding [41, 42] comprehensively integrated over 500 amino acid indices in AAIndex database and it gives a low-dimensional representation of these indices with in 19-dimensional. More details see Supplementary Information S1.

Unsupervised clustering and cluster-learning sampling In this work, two unsupervised clustering algorithms, K-means [19] and Louvain [21], were tested on CLADE. K-means clustering is computed using scikit-learn package with default kmeans++ initialization [46]. Louvain clustering is computed on a shared nearest neighbor graph implemented by Seurat package [48] (Supplementary Information S4).

The cluster-wise sampling probabilities depend on the average fitness of selected variants in each cluster. The cluster with higher average fitness has the higher probability to be selected. In k-th cluster at i-th hierarchy, the sampling probability is given by:

$$P_k^{(i)} = \frac{\frac{1}{\#C_k^{(i)}} \sum\limits_{j \text{ in } C_k^{(i)}} y_j}{\frac{1}{\sum\limits_l \#C_l^{(i)}} \sum\limits_l \sum\limits_{j \text{ in } C_l^{(i)}} y_j}, \tag{1}$$

where $C_l^{(i)} \subset I$ is the index set of l-th cluster at i-th hierarchy and I is the index set of the combinatorial library that gives each variant an unique index. And y_j is the fitness of j-th variant.

In deep hierarchical clustering, only K-means is applied since it is easy to control the number of clusters with a single hyperparameter K. For maximum hierarchy N, increment of clusters at i-th ($i \leq N$) hierarchy is given by K_i. The total number of clusters at maximum hierarchy is the sum of these numbers $\sum\limits_{i=1}^{N} K_i$. At a new hierarchy, clusters with higher average fitness are divided into more subclusters, and clusters with low average fitness are divided into fewer subclusters or not divided. The k-th parent cluster at $(i-1)$-th hierarchy will be divided into $L_k^{(i)}$ subclusters at i-th hierarchy, and $L_k^{(i)}$ is given by

$$L_k^{(i)} = \begin{cases} [P_k^{(i)} K_i] + 1, & \text{if } k \neq k_0 \\ K_i - \sum\limits_{j \neq k_0} [P_j^{(i)} K_i] + 1, & \text{if } k = k_0 \end{cases} \tag{2}$$

where k_0 is the cluster index such that this cluster has maximum average fitness from selected variants in all clusters and $[x]$ represents the largest integer not greater than x.

We summarize the flow of cluster-learning sampling together with required hyperparameters. The structure of clusters needs to be determined prior to the sampling process with $N+1$ hyperparameters, including maximum hierarchy N and the increment of clusters at each hierarchy K_i. The batch size, NUM_{batch}, is taken to be the number of variants being screened simultaneously in experiment. The batch size decides the

parameter	medium throughput	low throughput
NUM_{batch}	96	1
T	384	384
M	96	96
NUM_{1st}	96	96
$\text{NUM}_{hierarchy}$	96	96

Table 3: Numbers for simulated medium- and low-throughput systems in work.

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 experiment. During sampling, the first round selection selects NUM_{1st} variants, that are equally picked over clusters to have a rough coverage of all clusters. After the first-round selection, sampling probability is updated every batch according to Eq. (1), and a new hierarchy is generated after every $\text{NUM}_{hierarchy}$ variants is screened until reaching maximum hierarchy N. The sampling process generates NUM_{train} labeled variants to train the downstream supervised learning model. The top M variants predicted by CLADE are experimentally screened. These numbers, NUM_{1st}, $\text{NUM}_{hierarchy}$, NUM_{train}, and M are all required to be multiples of batch size NUM_{batch}. The $N+1$ hyperparameters for clustering were extensive explored in this work. Two sets of the other five hyperparameters were explored to simulate medium- and low-throughput systems (Table 3). In application, NUM_{batch} is picked according to experimental protocol and T can be picked according to screening capacity. The other three numbers can be selected according to our experiment and scaled to the suitable values.

Supervised learning The MLDE package [33] was used for the supervised learning model in this work. An ensemble of 16 regression models optimized by Bayesian hyperparameter optimizations were used. 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 picked and averaged to predict the fitness of unlabeled variants. Details see Supplementary Information S2 and Table S6-S7.

Evaluating metrics Various metrics were used to evaluate the training data diversity and CLADE outcome. *Mean fitness* and *max fitness* are calculated in three sets, including training data, the top M predicted variants and their union. *Global maximal fitness hitting rate* calculated the frequency that the global max fitness variant is successfully picked in multiple independent repeats. *Normalized discounted cumulative gain* (NDCG) is a measure of ranking quality to evaluate the predictive performance of CLADE on all unlabeled data. Its value is between 0 and 1. When NDCG is closed to 1, it indicates that variants ranked by the predicted fitness are similar to that ranked by the ground truth fitness. *Root mean square error* (RMSE) and *Pearson correlation* are used to evaluate the performance of the supervised learning for both cross validation and testing. *Modified functional attribute diversity* (MFAD) is a quantity to measure data diversity [49]. In this work, we used it to measure fitness and sequence diversity for training data. Suppose T is the training data size, MFAD is given by

$$\text{MFAD} = \frac{\sum_{i=1}^{T}\sum_{j=1}^{T} d_{ij}}{T}, \tag{3}$$

where d_{ij} represents the dissimilarity between i-th sample and j-th sample. For fitness diversity, the dissimilarity is calculated by the difference of fitness between two samples:

$$d_{ij}^{\text{fitness}} = |y_i - y_j| . \tag{4}$$

For sequence diversity, the dissimilarity is calculated by Euclidean distance between two samples of the physicochemical encoding:

$$d_{ij}^{\text{sequence}} = \|x_i - x_j\|_2 \tag{5}$$

where x_i is the physicochemical encoding feature vector of i-th variant, and $\|\cdot\|$ is the Euclidean distance.

Data Availability

The GB1 dataset [32] is an empirical fitness landscape for protein G domain B1 (GB1; PDB ID: 2GI9) binding to an antibody. The fitness was defined as the enrichment of folded protein bound to the antibody IgG-Fc. This data set contains 149,361 experimentally labeled variants out of 20^4=160,000 at four amino acid sites (V39, D40, G41, and V54). The fitness of the remaining 10,639 unlabeled variants is imputed, but they are not considered in this study. In this work, we linearly scaled the range of fitness to $[0, 1]$ by normalizing fitness to global maximum fitness.

In PhoQ dataset [39], a high-throughput assay for the signaling of the two-component regulatory system, PhoQ-PhoP sensor kinase and a response regulator (PDB ID: 3DGE), was developed with a YFP reporter expressed from a PhoP-dependent promoter. The combinatorial library was constructed at four sites (A284, V285, S288, and T289) for PhoQ. Phosphatase or kinase activity by stimulating PhoQ with high or low extracellular magnesium was performed. This two-step selection involving two libraries was used to select mutants that behaved similarly to the wild-type PhoQ. In this work, we took the data from the combinatorial library with high extracellular magnesium treatment, where it has large coverage with 140,517 quality-read variants out of 20^4=160,000. The fitness was defined as the enrichment of similar phosphatase activity with wild-type PhoQ to its substrate PhoP. We linearly scaled the range of fitness to $[0, 1]$ by normalizing fitness to global maximum fitness.

Code Availability

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

Supporting Information

Acknowledgments

This work was supported in part by NIH grants GM126189 and GM129004, NSF grants DMS-2052983, DMS-1761320, and IIS-1900473, NASA grant 80NSSC21M0023, Michigan Economic Development Corporation, Bristol-Myers Squibb 65109, and Pfizer. The authors thank The IBM TJ Watson Research Center, The COVID-19 High Performance Computing Consortium, NVIDIA, and MSU HPCC for computational assistance. The authors thank Frances Arnold's Lab for assistance with MLDE package and Michael T. Laub's Lab for assistance on PhoQ dataset.

Competing interests

The authors declare no competing interests.

References

[1] Kevin K Yang, Zachary Wu, and Frances H Arnold. Machine-learning-guided directed evolution for protein engineering. *Nature Methods*, 16(8):687–694, 2019.

[2] Niklas E Siedhoff, Ulrich Schwaneberg, and Mehdi D Davari. Machine learning-assisted enzyme engineering. *Methods in Enzymology*, 643:281–315, 2020.

[3] Harini Narayanan, Fabian Dingfelder, Alessandro Butté, Nikolai Lorenzen, Michael Sokolov, and Paolo Arosio. Machine learning for biologics: opportunities for protein engineering, developability, and formulation. *Trends in Pharmacological Sciences*, 2021.

[4] Stanislav Mazurenko, Zbynek Prokop, and Jiri Damborsky. Machine learning in enzyme engineering. *ACS Catalysis*, 10(2):1210–1223, 2019.

[5] Daniel Bojar and Martin Fussenegger. The role of protein engineering in biomedical applications of mammalian synthetic biology. *Small*, 16(27):1903093, 2020.

[6] Gi Bae Kim, Won Jun Kim, Hyun Uk Kim, and Sang Yup Lee. Machine learning applications in systems metabolic engineering. *Current Opinion in Biotechnology*, 64:1–9, 2020.

[7] Jian Tian, Ningfeng Wu, Xiaoyu Chu, and Yunliu Fan. Predicting changes in protein thermostability brought about by single-or multi-site mutations. *BMC Bioinformatics*, 11(1):1–9, 2010.

[8] Zixuan Cang and Guo-Wei Wei. Analysis and prediction of protein folding energy changes upon mutation by element specific persistent homology. *Bioinformatics*, 33(22):3549–3557, 2017.

[9] Lijun Quan, Qiang Lv, and Yang Zhang. STRUM: structure-based prediction of protein stability changes upon single-point mutation. *Bioinformatics*, 32(19):2936–2946, 2016.

[10] Sameer Khurana, Reda Rawi, Khalid Kunji, Gwo-Yu Chuang, Halima Bensmail, and Raghvendra Mall. Deepsol: a deep learning framework for sequence-based protein solubility prediction. *Bioinformatics*, 34(15):2605–2613, 2018.

[11] Brian Hie, Bryan D Bryson, and Bonnie Berger. Leveraging uncertainty in machine learning accelerates biological discovery and design. *Cell Systems*, 11(5):461–477, 2020.

[12] Menglun Wang, Zixuan Cang, and Guo-Wei Wei. A topology-based network tree for the prediction of protein–protein binding affinity changes following mutation. *Nature Machine Intelligence*, 2(2):116–123, 2020.

[13] Roshan Rao, Nicholas Bhattacharya, Neil Thomas, Yan Duan, Xi Chen, John Canny, Pieter Abbeel, and Yun S Song. Evaluating protein transfer learning with tape. *Advances in Neural Information Processing Systems*, 32:9689, 2019.

[14] Kevin K Yang, Zachary Wu, Claire N Bedbrook, and Frances H Arnold. Learned protein embeddings for machine learning. *Bioinformatics*, 34(15):2642–2648, 2018.

[15] Adam J Riesselman, John B Ingraham, and Debora S Marks. Deep generative models of genetic variation capture the effects of mutations. *Nature Methods*, 15(10):816–822, 2018.

[16] Ethan C Alley, Grigory Khimulya, Surojit Biswas, Mohammed AlQuraishi, and George M Church. Unified rational protein engineering with sequence-based deep representation learning. *Nature Methods*, 16(12):1315–1322, 2019.

[17] Tristan Bepler and Bonnie Berger. Learning protein sequence embeddings using information from structure. In *International Conference on Learning Representations*, 2018.

[18] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *arXiv preprint arXiv:1706.03762*, 2017.

[19] Greg Hamerly and Charles Elkan. Learning the k in k-means. *Advances in Neural Information Processing Systems*, 16:281–288, 2004.

[20] Brendan J Frey and Delbert Dueck. Clustering by passing messages between data points. *science*, 315 (5814):972–976, 2007.

[21] Vincent D Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: theory and experiment*, 2008(10): P10008, 2008.

[22] Erich Schubert, Jörg Sander, Martin Ester, Hans Peter Kriegel, and Xiaowei Xu. DBSCAN revisited, revisited: why and how you should (still) use DBSCAN. *ACM Transactions on Database Systems (TODS)*, 42(3):1–21, 2017.

[23] Yutong Sha, Shuxiong Wang, Peijie Zhou, and Qing Nie. Inference and multiscale model of epithelial-to-mesenchymal transition via single-cell transcriptomic data. *Nucleic Acids Research*, 48(17):9505–9520, 2020.

[24] Da Kuang, Chris Ding, and Haesun Park. Symmetric nonnegative matrix factorization for graph clustering. In *Proceedings of the 2012 SIAM international conference on data mining*, pages 106–117. SIAM, 2012.

[25] Sergio Oller-Moreno, Karin Kloiber, Pierre Machart, and Stefan Bonn. Algorithmic advances in machine learning for single cell expression analysis. *Current Opinion in Systems Biology*, 2021.

[26] Amit Saxena, Mukesh Prasad, Akshansh Gupta, Neha Bharill, Om Prakash Patel, Aruna Tiwari, Meng Joo Er, Weiping Ding, and Chin-Teng Lin. A review of clustering techniques and developments. *Neurocomputing*, 267:664–681, 2017.

[27] Yanfei Zhong, Ailong Ma, Yew soon Ong, Zexuan Zhu, and Liangpei Zhang. Computational intelligence in optical remote sensing image processing. *Applied Soft Computing*, 64:75–93, 2018.

[28] Olga Khersonsky Tawfik and Dan S. Enzyme promiscuity: a mechanistic and evolutionary perspective. *Annual review of biochemistry*, 79:471–505, 2010.

[29] Tyler N Starr and Joseph W Thornton. Epistasis in protein evolution. *Protein Science*, 25(7):1204–1218, 2016.

[30] Bruce J Wittmann, Yisong Yue, and Frances H Arnold. Machine learning-assisted directed evolution navigates a combinatorial epistatic fitness landscape with minimal screening burden. *bioRxiv*, 2020.

[31] Zachary Wu, SB Jennifer Kan, Russell D Lewis, Bruce J Wittmann, and Frances H Arnold. Machine learning-assisted directed protein evolution with combinatorial libraries. *Proceedings of the National Academy of Sciences*, 116(18):8852–8858, 2019.

[32] Nicholas C Wu, Lei Dai, C Anders Olson, James O Lloyd-Smith, and Ren Sun. Adaptation in protein fitness landscapes is facilitated by indirect paths. *Elife*, 5:e16965, 2016.

[33] Bruce J Wittmann, Kadina E Johnston, Zachary Wu, and Frances H Arnold. Advances in machine learning for directed evolution. *Current Opinion in Structural Biology*, 69:11–18, 2021.

[34] Guangyue Li, Yijie Dong, and Manfred T Reetz. Can machine learning revolutionize directed evolution of selective enzymes? *Advanced Synthesis & Catalysis*, 361(11):2377–2386, 2019.

[35] Yutaka Saito, Misaki Oikawa, Hikaru Nakazawa, Teppei Niide, Tomoshi Kameda, Koji Tsuda, and Mitsuo Umetsu. Machine-learning-guided mutagenesis for directed evolution of fluorescent proteins. *ACS synthetic biology*, 7(9):2014–2022, 2018.

[36] Claire N Bedbrook, Kevin K Yang, Austin J Rice, Viviana Gradinaru, and Frances H Arnold. Machine learning to design integral membrane channelrhodopsins for efficient eukaryotic expression and plasma membrane localization. *PLoS computational biology*, 13(10):e1005786, 2017.

[37] Philip A Romero, Andreas Krause, and Frances H Arnold. Navigating the protein fitness landscape with gaussian processes. *Proceedings of the National Academy of Sciences*, 110(3):E193–E201, 2013.

[38] Derek M Mason, Simon Friedensohn, Cédric R Weber, Christian Jordi, Bastian Wagner, Simon Meng, Pablo Gainza, Bruno E Correia, and Sai T Reddy. Deep learning enables therapeutic antibody optimization in mammalian cells by deciphering high-dimensional protein sequence space. *BioRxiv*, page 617860, 2019.

[39] Anna I Podgornaia and Michael T Laub. Pervasive degeneracy and epistasis in a protein-protein interface. *Science*, 347(6222):673–677, 2015.

[40] Shuichi Kawashima, Hiroyuki Ogata, and Minoru Kanehisa. AAindex: amino acid index database. *Nucleic Acids Research*, 27(1):368–369, 1999.

[41] Dan Ofer and Michal Linial. ProFET: Feature engineering captures high-level protein functions. *Bioinformatics*, 31(21):3429–3436, 2015.

[42] Alexander G Georgiev. Interpretable numerical descriptors of amino acid space. *Journal of Computational Biology*, 16(5):703–723, 2009.

[43] Swagata Pahari, Gen Li, Adithya Krishna Murthy, Siqi Liang, Robert Fragoza, Haiyuan Yu, and Emil Alexov. SAAMBE-3D: Predicting effect of mutations on protein–protein interactions. *International Journal of Molecular Sciences*, 21(7):2563, 2020.

[44] Alexander Rives, Joshua Meier, Tom Sercu, Siddharth Goyal, Zeming Lin, Jason Liu, Demi Guo, Myle Ott, C Lawrence Zitnick, Jerry Ma, et al. Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences. *Proceedings of the National Academy of Sciences*, 118(15), 2021.

[45] Claire Strain-Damerell and Nicola A Burgess-Brown. High-throughput site-directed mutagenesis. In *High-Throughput Protein Production and Purification*, pages 281–296. Springer, 2019.

[46] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. Scikit-learn: Machine learning in Python. *the Journal of machine Learning research*, 12:2825–2830, 2011.

[47] AA Zamyatnin. Protein volume in solution. *Progress in Biophysics and Molecular Biology*, 24:107–123, 1972.

[48] Andrew Butler, Paul Hoffman, Peter Smibert, Efthymia Papalexi, and Rahul Satija. Integrating single-cell transcriptomic data across different conditions, technologies, and species. *Nature Biotechnology*, 36 (5):411–420, 2018.

[49] Dénes Schmera, Tibor Erős, and János Podani. A measure for assessing functional diversity in ecological communities. *Aquatic Ecology*, 43(1):157–167, 2009.

Supplementary Files

This is a list of supplementary files associated with this preprint. Click to download.

- supplement6.pdf
- flatWeispc.pdf
- flatWeiepc.pdf